Supercontinuum pulse characterisation and compression

by

Ruan Viljoen

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science in the Faculty of Science at Stellenbosch University

Supervisor: Prof. E.G. Rohwer

Co-supervisor: Dr. P. Neethling

March 2017

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: ..... March 2017 .....

Abstract

This project centers around the development of infrastructure, specifically a broadband white light source, for integration into non-linear microscopes. The broadband light source needed for such an instrument comprises an all-normal dispersion photonic crystal fibre (ANDi PCF) (which produces a supercontinuum) and a pulse compressor (which produces the needed ultra short pulses).

Polarisation and power characteristics of the ANDi PCF are investigated in order to establish the optimal working conditions with the broadest spectral output coinciding with the highest power throughput. The bandwidth can be increased from about 12 nm to 200 nm (depending on the pump power) and enables us to decrease the pump pulse length from about 80 fs to below 10 fs if we are able to fully compress the full bandwidth.

A particularilly suitable pulse characterisation technique, multiphoton intrapulse interference phase scan (MIIPS), is used to characterise and compress the supercontinuum pulses produced by the PCF. The efficacy of this technique is computationally and experimentally tested through computer simulations and laboratory experiments. From the simulations we see that the MIIPS technique is able to adequately reconstruct the phase of the supercontinuum pulses which can then be used to compress the pulses.

Experiments performed on bandwidths around $\sim$ 80 nm, show that the MIIPS algorithm is able to reconstruct the supercontinuum pulse's phase and in the process compress it. A measurement of the GVD of a SF6 glass cube with known dispersion is used as supplementary evidence of the effectiveness of the MIIPS technique.

A SLM assisted IFROG measurement is performed on the compressed pulses to further test the degree of compression after MIIPS. This measurement shows a decrease in pulse length from picosecond regime to $\sim$ 23 fs. This is near to the Fourier limit of $\sim$ 12 fs for these bandwidths.

Opsomming

In hierdie projek word die infrastruktuur van 'n hoë-resolusie nie-lineêre mikroskoop ontwikkel. Spesifiek word daar gekyk na generering van breëband lig deur middel van 'n fotoniese-kristalvesel (PCF) en 'n puls kompressor.

Polarisasie en drywing karaktereienskappe van die PCF word ondersoek om optimale werks omstandighede vas te stel. Vir hierdie optimale werks omstandighede benodig ons breëband lig met die hoogste drywing deursette. Afhangend van die pomp drywing, kan die bandwydte van die lig van 12 nm tot sowat 200 nm verbreed. Dit stel ons instaat om die pomp pulslengte van 80 fs na onder 10 fs saam te pers.

Vir die doeleindes van die projek is die 'multiphoton intrapulse interference phase scan' of MIIPS die ideale metode om ons breëband pulse te karakteriseer en saam te pers. Ons wys dat ons die toepaslikheid van hierdie metode numeries en eksperimenteel getoets het. Ons het vasgestel vanaf ons simulasies dat ons in staat is om die fase verhoudings van spektrale komponente vas te stel.

Eksperimente wat uitgevoer is met lig met 'n breëband van rondom 80 nm wys dat die MIIPS algoritme in staat is om die superkontinue pulse se fase vas te stel en in die proses die pulse saam te pers. 'n Meting van die GVD van 'n blokkie SF6 glas met 'n bekende dispersie is gebruik as bykomende inligting om die effektiwiteit van MIIPS te illustreer.

'n SLM IFROG meting is uitgevoer op die kort pulse as 'n toets hoe kort die pulse is na 'n MIIPS meting. Hierdie metode het gewys dat die pulse lengte van die pikosekonde regime na $\sim$ 23 fs saam gepers is. Hierdie pulse is naby aan die Fourier limiet van $\sim$ 12 fs vir hierdie bandwydte.

Acknowledgements

I would like to express my sincere gratitude to the following people:

- My supervisors Prof Erich Rohwer and Dr Pieter Neethling, for providing excellent supervision during my project and always being available to help when it was needed.
- Dr Dirk Spangenberg for mentoring me throughout my MSc.
- Prof Heinrich Schwoerer for generously providing me with funding.
- My friends and family, especially my parents, for supporting me throughout my academic career.

Contents

List of Figures

Chapter 1

Introduction

Studying objects smaller than what the eye can see has been a field of interest to humans from before the first microscope was invented. As time progressed more complex methods of studying smaller and smaller objects were invented and has allowed science the means to describe on different scales the building blocks of the universe. Recent developments in the field of microscopy has seen the implementation of supercontinuum (white) light generated by fibres in microscopies such as multiphoton microscopy, reflectance confocal microscopy and optical coherence tomography which allows for new and interesting ways of studying objects on the microscale with high resolution [4].

Working towards such high resolution microscopy systems based on white light, one has to develop the infrastructure (a pulse compressor of sorts), to generate ultrashort supercontinuum pulses which is essential for these microscopies.

The main underlying principle for such infrastructure lies in the inverse nature of light pulses in time and frequency and must be considered in the design of this system. Simply put: Moving past the Fourier limit and decreasing pulse length requires a larger spectral bandwidth.

In this study supercontinuum pulses are produced by an all-normal dispersion photonic crystal fibre (ANDi PCF) pumped with 80 fs pulses with a spectral bandwidth of about 20 nm and a repetition rate around 80 MHz. Through nonlinear processes in the PCF the bandwidth of the input pulse is increased. The bandwidth of the supercontinuum is determined by the input power, but this method of creating supercontinuum pulses has a side affect which requires some compensation.

While the pulses propagate through the fibre it not only undergoes an increase in spectral bandwidth, but is also stretched out in time due to the physical properties of the fibre's structure and material. These stretched out supercontinuum pulses can be characterised and compressed by determining the temporal relationship between the spectral components of the pulse (the spectral phase).

A 4f-shaper in combination with a one dimensional spatial light modulator (1D SLM) can be used to characterise the pulses as well as compress them. The 4f-shaper essentially

disperses the supercontinuum and maps the spectral components to spatial coordinates and then recombines them to form a pulse again. Adding a SLM to the system allows for arbitrary shaping of the amplitude and phase of these spectral components. This opens up the possibility to reverse the phase distortions caused by the PCF and produce supercontinuum pulses with pulse durations shorter than that of the pump pulses.

Before one can do this however, one has to determine the phase of the pulses. Many phase retrieval techniques exist, but a particularly suitable technique is the multiphoton intrapulse interference phase scan (MIIPS). This techniques is implemented through the use of the 4f-shaper-SLM combination and retrieves the phase and simultaneously correct for the phase to compress the pulse. This technique exploits the relationship between the intensity of second harmonic generated light (SHG) in a crystal and the phase of an input pulse, and furthermore, compresses the pulses where the SHG light is produced.

The different components comprising the so called pulse compressor is studied and characterised. The MIIPS technique is experimentally and numerically studied and implemented in a working setup.

Chapter 2

Description of short pulses

In order to manipulate pulses we have to develop mathematical tools to describe pulses of electromagnetic radiation. In this chapter these tools are developed, followed by a discussion on how short pulses are typically measured and characterized.

Ultrashort laser pulses are electromagnetic pulses (wave packets) and are thus described by the time and space dependent electric field [5]. Maxwell's equations govern the interaction with materials and hence the propagation of electromagnetic waves through materials.

2.1 Monochromatic electromagnetic radiation

For electromagnetic waves in vacuum there will be no medium to polarise and no magnetisation (i.e. $\vec{P} = \vec{M} = 0$). Applying this to the general representation of Maxwell's equations produce [6]:

$$\begin{cases} \vec{\nabla} \cdot \vec{E} = 0 \\ \vec{\nabla} \times \vec{E} = -\dfrac{\partial \vec{B}}{\partial t} \\ \vec{\nabla} \cdot \vec{B} = 0 \\ \vec{\nabla} \times \vec{B} = \mu_0 \epsilon_0 \dfrac{\partial \vec{E}}{\partial t} \end{cases}, \tag{2.1}$$

where $\vec{E}$ and $\vec{B}$ represent the electric and magnetic fields respectively, and μ_0 and ϵ_0 represent the permiability and permitivity of space. Decoupling these first order partial derivatives produce the electric and magnetic wave equations - describing the propagation of light. At first glance it would seem that the magnetic field is an order of c (the speed of light in vacuum) smaller than the electric field, when considering the relationship $Bc = E$. This, however, is only a way to compute the one from the other, while further investigation shows that the energy contained within these fields are in fact equal [6].

It is common to only consider the electric field component of the electromagnetic radiation, as the media under consideration is not magnetic. If we then know the electric field, then we are able to calculate the magnetic field since we know that they are directly proportional. Consider the wave equation for the electric field:

$$\nabla^2 \vec{E} = \mu_0 \epsilon_0 \frac{\partial^2 \vec{E}}{\partial t^2}. \tag{2.2}$$

The speed of light in vacuum is then defined as $c = \frac{1}{\sqrt{\mu_0 \epsilon_0}}$. Equation (2.2) is a second-order partial differential equation and has a solution of the following form:

$$\tilde{\vec{E}}(z,t) = \tilde{\vec{E}}_0 \exp\left[i(\omega t - kz - \phi)\right] \tag{2.3}$$

$$= \tilde{\vec{E}}_0 \cos\left(\omega t - kz - \phi\right) + i\tilde{\vec{E}}_0 \sin\left(\omega t - kz - \phi\right), \tag{2.4}$$

where

$$\tilde{\vec{E}}_0 = \tilde{E}_0 \hat{n}. \tag{2.5}$$

Here $\hat{n}$ is the polarisation vector, which is perpindicular to the direction of propagation. Equation (2.3) describes a transverse, monochromatic plane wave propagating in the z direction with angular frequency ω and wave number, $|\vec{k}| = \frac{\omega}{c} = \frac{2\pi}{\lambda}$. We note that the physical electric field is the real part of this solution. The complex representation in equation (2.3) is merely for mathematical convenience, as it allows to easily keep track of the phase component. This representation can be further generalised to:

$$\tilde{\vec{E}}(\vec{r},t) = \tilde{E}_0 \exp\left[i(\omega t - \vec{k}\cdot\vec{r} - \phi)\right]\hat{n}. \tag{2.6}$$

The physical electric field is retrieved by taking the real part:

$$\vec{E}(\vec{r},t) = \Re[\tilde{\vec{E}}(\vec{r},t)] = E_0 \cos\left(\omega t - \vec{k}\cdot\vec{r} - \phi\right)\hat{n}. \tag{2.7}$$

2.2 Constructing pulses

We consider the electric field at a given point in space:

$$E(t) = \Re[E_0 e^{i\omega_0 t}]. \tag{2.8}$$

This is still a plane wave that stretches out infinitely in time. This wave can be fully described in either the temporal or frequency domain. To switch between these two domains the complex Fourier or Inverse Fourier Transform is applied [5]:

$$E(\omega) = \mathcal{F}\{\varepsilon(t)\} = \int_{-\infty}^{\infty} \varepsilon(t) e^{-i\omega t} dt = |E(\omega)| e^{-i\varphi(\omega)} \tag{2.9a}$$

$$\varepsilon(t) = \mathcal{F}^{-1}\{E(\omega)\} = \frac{1}{2\pi}\int_{-\infty}^{\infty} E(\omega) e^{i\omega t} d\omega. \tag{2.9b}$$

Here $\varepsilon(t)$, $\phi(\omega)$ and $|E(\omega)|$ represent a pulsed electric field, the spectral phase and spectral amplitude respectively. For equation (2.8) the Fourier Transform to the spectral domain will produce a Dirac Delta function ($\delta(\omega - \omega_0)$) at ω_0 (the entire spectral content of the wave). As Hirlimann [7] points out, this monochromatic plane wave is the 'absolute opposite' of a pulse. So one can decrease the spectral content and there will obviously be no electric field, but what happens when the spectral content is increased?

For illustrative purposes consider a spectrum with a Gaussian shape centred at ω_0, now the central frequency, with a spectral width relating to the form factor Γ (which relates to the standard deviation ,σ). A Gaussian shape is an easy choice as the Fourier transform of this function is well known to also be Gaussian:

$$E(\omega) = e^{-\frac{(\omega-\omega_0)^2}{4\Gamma}}, \qquad \Gamma = \frac{\sigma^2}{2}. \tag{2.10}$$

Applying the inverse Fourier transform to 2.10 gives the time equivalent:

$$\begin{aligned} \varepsilon(t) &= \frac{1}{2\pi}\int_{-\infty}^{\infty} E(\omega)e^{i\omega t}d\omega \\ &= \sqrt{2\Gamma}e^{-\Gamma t^2}e^{-i\omega_o t}. \end{aligned} \tag{2.11}$$

With the real electric field as:

$$E(t) = \Re[\varepsilon(t)]. \tag{2.12}$$

We see that the original electric field is present but multiplied with a Gaussian function. This is a Gaussian amplitude modulation of a wave with a centre frequency at ω_0 resulting in a pulse.

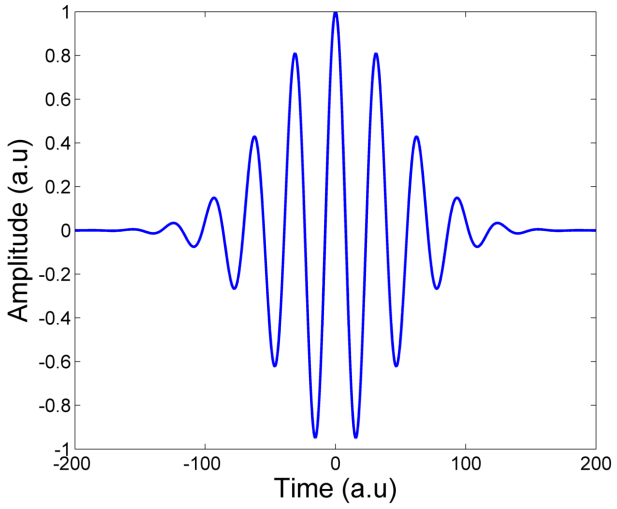


Figure 2.1: Simulated arbitrary electric field with Gaussian amplitude modulation representing a pulse as described in equation (2.11).

The general expression for the temporal dependence of the electric field pulse can be expressed as [8]:

$$\varepsilon(t) = \frac{1}{2}\sqrt{I(t)}e^{i(\omega_0 t - \phi(t))} + c.c. \tag{2.13}$$

Here $e^{i\omega_0 t}$ represents the rapidly varying carrier wave with ω_o as the carrier frequency; $I(t)$ and $\phi(t)$ represent the intensity and phase of the pulse respectively; and *c.c.* refers to the complex conjugate. Also note that under the scalar approximation, that this field is linearly polarised. Further simplification of this expression is done by performing the analytic signal approximation. Under this approximation $I(t)$ and $\phi(t)$ are assumed to vary slowly compared to $e^{i\omega_0 t}$:

$$\varepsilon_0(t) = \sqrt{I(t)}e^{-i\phi(t)}. \tag{2.14}$$

It should be clear that equation (2.13) has simply been multiplied by 2 and been removed of the complex conjugate term and the rapidly varying carier wave. Equation (2.14) is the complex amplitude of the wave - or the envelope of the rapidly varying electric field. Using this definition greatly simplifies the representation of the electric field and is merely for convenience.

In the frequency domain however, the Fourier transform of equation (2.13) is generally preferred:

$$E(\omega) = \sqrt{S(\omega)}e^{-i\varphi(\omega)}. \tag{2.15}$$

where $S(\omega)$ represents the spectrum and $\varphi(\omega)$ the spectral phase. This representation allows for the spectrum to be centred at the carrier frequency ω_0. Constructing the spectral field in this form is convenient as these two quantities (spectrum and phase) are the quantities that will be measured.

2.3 Pulse propagation

Consider again the Gaussian pulse, equation(2.10), now propagating through a transparent medium of length L with a refractive index of $n(\omega)$ and an absorption coefficient, $\alpha(\omega)$. Propagating through this medium will produce a new pulse:

$$E_{out}(\omega, L) = E(\omega)e^{(-i\frac{\omega n(\omega)}{c}L)}e^{(-\alpha(\omega)L/2)}. \tag{2.16}$$

This modifies the spectrum and spectral phase as follows:

$$S_{out}(\omega) = S(\omega)e^{(-\alpha(\omega)L)} \tag{2.17a}$$

$$\varphi_{out}(\omega) = \varphi(\omega) + \frac{\omega n(\omega)}{c}L. \tag{2.17b}$$

One can think of the spectral phase as an account keeping tool that keeps track of the spectral components relative to each other. The physical cause of the change in spectral phase can be attributed to spectral components moving at different speeds through the medium depending on the index of refraction. While propagating through the medium, these components will spread out and away from each other, changing the spectral phase in the process and also changing how the spectral components add up to form the pulse.

We define the frequency-dependant propagation factor as $k(\omega) = \frac{\omega n(\omega)}{c}$ and apply the Taylor expansion:

$$k(\omega) = k(\omega_0) + \dot{k}(\omega - \omega_0) + \frac{\ddot{k}}{2}(\omega - \omega_0)^2 + ... \tag{2.18}$$

where

$$\dot{k} = (\frac{dk(\omega)}{d\omega})_{\omega_0} \tag{2.19a}$$

and

$$\ddot{k} = (\frac{d^2k(\omega)}{d\omega^2})_{\omega_0}. \tag{2.19b}$$

Substituting this into equation (2.16), neglecting the absorption due to propagating through a medium, and taking the Fourier transform will produce the temporal pulse after some propagation L [7]:

$$\varepsilon(t, L) = \sqrt{\frac{\Gamma(L)}{\pi}} \exp\left[i\omega_0\Big(t - \frac{L}{v_\phi(\omega_0)}\Big) - \Gamma(L)\Big(t - \frac{L}{v_g(\omega_0)}\Big)^2\right]. \tag{2.20}$$

With the following definitions:

$$v_\phi(\omega_0) = (\frac{\omega}{k})_{\omega_0} \tag{2.21a}$$

$$v_g(\omega_0) = ((\dot{k})^{-1})_{\omega_0} \tag{2.21b}$$

$$\frac{1}{\Gamma(L)} = \frac{1}{\Gamma} + 2i\ddot{k}L. \tag{2.21c}$$

We can neglect the absorption as it only has an influence on the amplitude.

Equation (2.21a) is referred to as the phase velocity, v_ϕ, and is the speed of the plane wave components of the pulse in the medium. This quantity can interestingly exceed that of the speed of light, but carries no information. Information is, however, carried in the group velocity, v_g, or the velocity with which the envelope of the pulse propagates through the medium.

Taking a derivative of the inverse of the group velocity will produce a quantity referred to as the group velocity dispersion (GVD):

$$GVD = \frac{\partial^2 k}{\partial \omega_0^2}. \tag{2.22}$$

The GVD is then equivalently the second derivative of the propagation factor - $\ddot{k}$ - and is present in the equation (2.21c). The form factor of the Gaussian propagating through a medium, with $\ddot{k}$ and L greater than zero, will be greater than the original form factor. This means that the GVD will have a dispersive effect on the envelope and stretch it out in time, which can be shown by deriving the FWHM of the Gaussian pulse and comparing it to the unchirped pulse[1].

[1] This derivation is done in section 2.5.

A pulse travelling through a medium will have a slower speed compared to travelling through vacuum. The time the pulse take to traverse the medium is known as group delay and can be defined as the inverse group velocity per unit length, and has units of time, or is the first derivative of the spectral phase with respect to frequency:

$$t_g = \frac{L}{v_g} = \frac{\partial \varphi}{\partial \omega}. \tag{2.23}$$

Taking the derivative of the group delay with respect to frequency will produce the group delay dispersion (GDD), which is the frequency dependence of the group delay:

$$GDD = \frac{\partial t_g}{\partial \omega} = \frac{\partial^2 \varphi}{\partial \omega^2}. \tag{2.24}$$

GDD is also in units of time and is usually presented in fs^2. Note that the group velocity dispersion is the group delay dispersion per unit length.

2.4 Spectral Phase

For the Gaussian envelope, we considered the frequency components to be well behaved and in phase with respect to each other. This translates to the spectral phase term, $\varphi(\omega)$, being equal to zero (also referred to as flat phase). This quantity, the spectral phase, contains time vs. frequency information or in other words how the frequency components are separated in time[8]. The Taylor series of the spectral phase is presented:

$$\varphi(\omega) = \varphi(\omega_0) + \dot{\varphi}(\omega_0)(\omega - \omega_0) + \frac{\ddot{\varphi}(\omega_0)}{2}(\omega - \omega_0)^2 + ... \tag{2.25}$$

The first term in equation (2.25), $\varphi(\omega_0)$, represent the zeroth-order phase or the absolute phase. This term represents the relative phase between the carrier wave and the carrier envelope. For pulses with large spectral content this term is not of major importance.

The second term, $\dot{\varphi}(\omega_0)$, is referred to as the first-order spectral phase, which corresponds to a shift in time. By performing a Fourier transform of a spectrum with some arbitrary first-order phase added, we can illustrate this temporal shift. Figure (2.2) serves as an example of the effect that this term has on the temporal pulse.

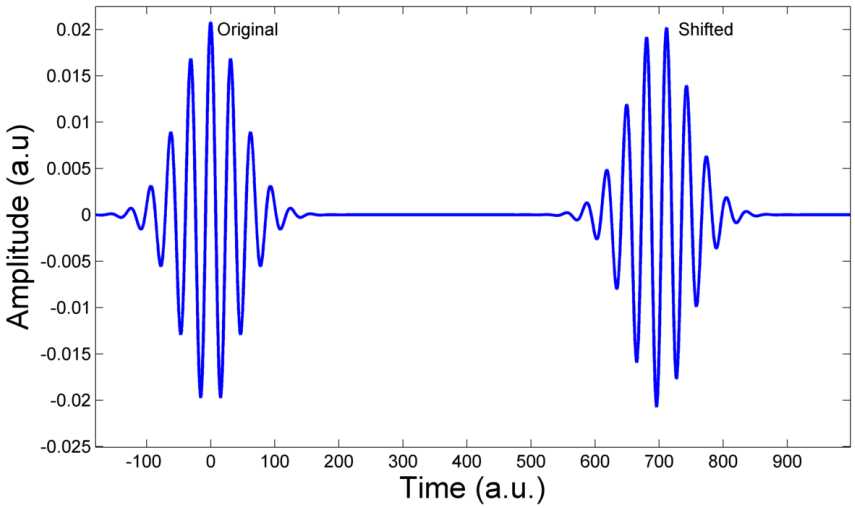


Figure 2.2: Simulated arbitrary pulse together with pulse with added first-order phase clearly shifted backwards in time.

First-order phase is generally neglected because measurement techniques are not generally sensitive as to when a pulse arrives. It can however be useful to exploit the nature of this term in some correlation techniques [2].

Second-order (quadratic) phase, $\ddot{\varphi}$, corresponds to the third term in equation (2.25). A non zero value of quadratic phase causes the pulse to have red frequencies leading the pulse with blue frequencies trailing behind. This is referred to as a chirp [7].

Higher order phase terms further contribute to the phase distortions of the pulse and can lead to intricate temporal pulse shapes [8].

2.5 Pulse duration

For most common well-behaved pulses, both the spectral and temporal fields can be described by the full width half maximum, $\Delta\omega_p$, and temporal pulse duration, $\Delta\tau_p$.

Specifically for our Gaussian spectrum the spectral width is related to the form factor in the following way:

$$\Delta\omega_p = 2\sqrt{2\ln 2}\sqrt{2\Gamma}. \tag{2.26}$$

With the Fourier transform 'connecting' the spectral and temporal characteristics of the electric field, implies that these quantities cannot vary independently of each other. It can be shown that they are governed by the following relationship (a type of uncertainty principle):

$$\Delta\omega_p\Delta\tau_p \geq 2\pi TBP. \tag{2.27}$$

Where the time bandwidth product (TBP) is a numerical constant related to the shape of the pulse. For a Gaussian pulse this value is $\sim 0{,}441$. One can find a list of these values

[2] This is exactly what is done in a spatial light modulator (SLM) assisted interferometric frequency resolved optical gating (IFROG), which will be discussed later.

for specific pulse shapes in most literature that discuss pulses [9]. When this product is equal to this value then the pulse is Fourier or Transform limited. This means the pulse duration is the shortest it can be with it's corresponding frequency bandwidth. Further increasing the bandwidth allows for further shortening of pulses.

We derive an expression for the pulse duration of a Gaussian pulse by considering the general case of a Gaussian pulse propagating through some material with length L, equation (2.20). In order to derive the expression for the FWHM of the pulse one has to calculate the optical intensity of the pulse:

$$I(t) = \frac{1}{2}\epsilon_0 cn|\varepsilon(t,L)|^2. \tag{2.28}$$

To simplify this calculation one can rewrite $\Gamma(L)$ in form:

$$\Gamma(L) = \frac{\Gamma}{1+\xi^2L^2} - i\frac{\Gamma\xi L}{1+\xi^2L^2}. \tag{2.29}$$

where $\xi = 2\Gamma\ddot{k}$. One can subsequently expand the exponential part of equation (2.20):

$$\exp\Big[-\frac{\Gamma}{1+\xi^2L^2}\Big(t-\frac{L}{v_g}\Big)^2 + i\frac{\Gamma\xi L}{1+\xi^2L^2}\Big(t-\frac{L}{v_g}\Big)^2\Big]. \tag{2.30}$$

The intensity is then calculated to be:

$$I(t) = \frac{1}{2}\epsilon cn\sqrt{\frac{\Gamma(L)^*\Gamma(L)}{\pi^2}}\exp\Big[-\frac{2\Gamma}{1+\xi^2L^2}\Big(t-\frac{L}{v_g}\Big)\Big]. \tag{2.31}$$

The maximum intensity for this pulse can be found where $t = \frac{L}{v_g}$. The FWHM is then calculated by calculating at what times the intensity will be half this value. Setting equation (2.31) equal to $\frac{1}{2}I(\frac{L}{v_g})$ and solving for t produces:

$$t = \frac{2L}{v_g} \pm \frac{\sqrt{2\ln 2}\sqrt{\Gamma(\xi^2L^2+1)}}{2\Gamma}. \tag{2.32}$$

It follows that the FWHM is calculated with:

$$FWHM = t_+ - t_-. \tag{2.33}$$

The general expression for the FWHM of a Gaussian pulse propagating through a medium with a refractive index $n(\omega) > 1$ of length L is then:

$$\Delta\tau_p = \sqrt{\frac{2\ln 2}{\Gamma}}\sqrt{\xi^2L^2+1}. \tag{2.34}$$

For a pulse propagating in vacuum, i.e. $n(\omega) = 1$, $\ddot{k}$ will be zero and the FWHM of the pulse will be:

$$\Delta\tau_p = \sqrt{\frac{2\ln 2}{\Gamma}}. \tag{2.35}$$

Comparing equations (2.34) and (2.35) one can see that a pulse propagating through a medium with some refractive index larger than 1 will have a larger pulse length after some distance compared to the same pulse propagating in a vacuum.

2.6 Ultrashort pulse measurement

Measuring of pulses can be directly performed for pulses down to the picosecond regime. Unfortunately conventional measuring methods are unable to temporally resolve femtosecond and shorter pulses. Ingenious methods have been developed in order to measure such pulses of which one is the interferometric autocorrelation (IAC). The IAC is able to accurately determine the temporal pulse duration and temporal phase modulations [10].

As the name suggests, this technique is the interference of a pulse with itself for different time delays. Such an autocorrelation signal is described by Diels et al [11]:

$$G^{(2)}(\tau) = \int_{-\infty}^{\infty} |[\varepsilon(t) + \varepsilon(t-\tau)]^2|^2 dt. \tag{2.36}$$

Here τ represents the time delay between separated pulses described by ε. Expanding this equation produces:

$$G^{(2)}(\tau) = C_0(\tau) + Re[4C_1(\tau)e^{i\omega_0\tau}] + Re[2C_2(\tau)e^{2i\omega_0\tau}], \tag{2.37}$$

where

$$C_0(\tau) = \int_{-\infty}^{\infty} [E^4(t-\tau) + E^4(t) + 4E^2(t-\tau)E^2(t)]dt \tag{2.38}$$

$$C_1(\tau) = \int_{-\infty}^{\infty} [E(t-\tau)E(t)[E^2(t-\tau) + E^2(t)]e^{i[\phi(t-\tau)-\phi(t)]}]dt \tag{2.39}$$

$$C_2(\tau) = \int_{-\infty}^{\infty} [E^2(t-\tau)E^2(t)e^{2i[\phi(t-\tau)-\phi(t)]}]dt. \tag{2.40}$$

Substituting $\tau = 0$, where there is no time delay between pulses, and $\tau = \infty$, where pulses do not overlap in time, into equation (2.36) results in:

$$G^{(2)}(0) = 16 \int_{-\infty}^{\infty} E^4(t)dt \tag{2.41a}$$

$$G^{(2)}(\infty) = 2 \int_{-\infty}^{\infty} E^4(t)dt. \tag{2.41b}$$

This gives rise to a constant background to signal ratio of 1:8 as illustrated in figure(2.3). Another attribute of the IAC signal is that it is symmetric, implying that pulses cannot be uniquely characterised with this method. If one is interested in determining the spectral intensity and spectral phase of a pulse, other techniques need to be employed.

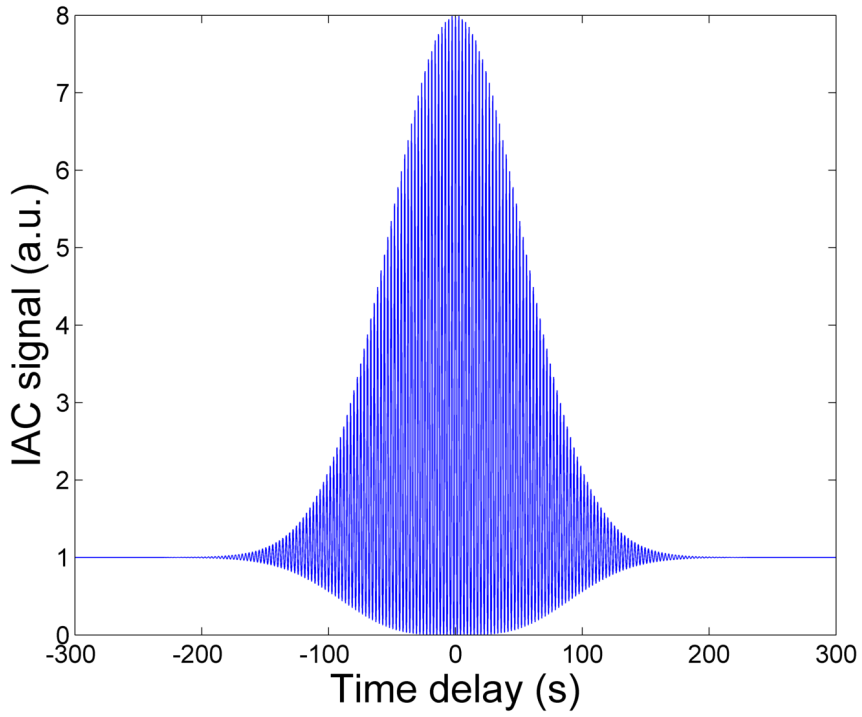


Figure 2.3: Simulated autocorrelation of laser pulse with centre wavelength of 800 nm and temporal width of 80 fs. Note the background to signal ratio of 1:8 seen at -300 s and 0 s respectively.

In order to extract the FWHM from the measurement one has to assume a pulse shape, which will determine the conversion factor. Assuming a Gaussian pulse shape the conversion is simply $\Delta\tau_{IAC} = \Delta\tau \times \sqrt{2}$.

Chapter 3

Supercontinuum: generation, characterisation and compression

Increasing the frequency bandwidth of laser pulses is crucial in generating shorter laser pulses. Broadband laser light, also referred to as a supercontinuum (SC), can be generated in bulk media, conventional fibres and photonic crystal fibres (PCFs) [12].

In deciding which of these methods to use in generating supercontinuum pulses, the characteristics of the supercontinua produced by each must be considered. The most important characteristic is the coherence of the supercontinuum. In other words the produced spectral components should have a fixed phase relationship. Coherent supercontinuum pulses are compressible and thus desirable to use in a pulse compressor.

3.1 Photonic crystal fibre

3.1.1 All-normal dispersion fibre

A photonic crystal fibre (PCF) is a highly ordered fibre optic waveguide capable of increasing the bandwidth of an incident pulse. PCF's are fabricated by stacking a preform (a macro structure to be heated and pulled to form a fibre) with the same structure desired in the microstructure [13]. In this case the preform is stacked in such a way that a solid core is surround by a hexagonal airhole cladding. This highly ordered crystal-like structure is what gives the PCF its name. After pulling the PCF preform the core will have a diameter in the order of micro meters. Having an airhole structure around the core creates a large index of refraction contrast which allows light to be guided along this core by means of total internal reflection.

The structure of such a PCF can be characterised in terms of two parameters: pitch (Λ) and hole diameter (d) which is expressed as the relative hole size (d/Λ) as illustrated in figure (3.1). Adjusting these parameters changes the effective dispersion profile of the fibre. This means that in combination with the known dispersion profile of a

material one is able to tailor a PCF's dispersion profile with different characteristics.

If one chooses the correct values for Λ and d/Λ, one can fabricate a fibre with the entire dispersion profile below zero and thus have an all-normal dispersion (ANDi) photonic crystal fibre [14]. The main mechanism behind the supercontinuum generation in such a PCF is self phase modulation with soliton fission suppressed. These mechanisms are fully described and discussed in detail elsewhere [15], and is not crucial in the understanding of implementation of an ANDi PCF. The ability to tailor the characteristics of a PCF makes it particularly useful for creating coherent broadband light sources.

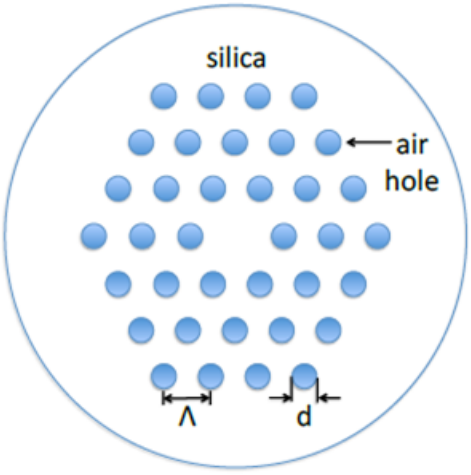


Figure 3.1: Schematic representation of internal PCF structure [1].

Changing the pump pulse parameters (pump power and pulse length) influences the supercontinuum pulse output [15]. Increasing the power of the pump pulse increases the spectral bandwidth as well as the temporal pulse length of the output pulse. Increased bandwidth implies that shorter pulse durations can be achieved by temporal pulse compression.

3.1.2 Simulation of temporal profile

In depth studies on the behaviour of ANDi PCF's by A. Heidt [15] have yielded accurate numerical simulations that produce complete representations of the output from such fibers. That is, the temporal profile (containing temporal phase information) which can be easily Fourier transformed to access the spectral phase and amplitude of the pulse. The following spectra are from simulations, provided by A. Heidt, for a 15 cm long ANDi PCF with input powers of 50mW, 100mW, 150mW and 200mW respectively [16]. With these pulse simulations one is able to test the viability of phase retrieval schemes computationally before implementing it in a real experiment.

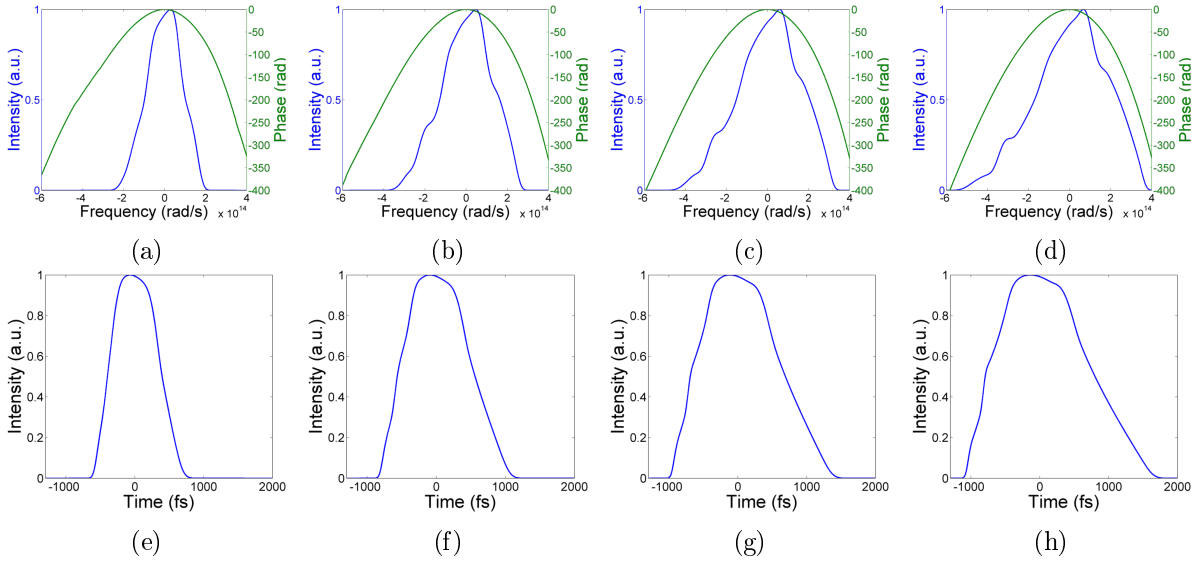


Figure 3.2: Presented in figures (a-d): Relative frequency vs. spectrum (blue) and spectral phase (green) with increasing spectral bandwidth from a to d; (e-h) Temporal profiles corresponding to spectra in (a-d)

Comparing figures (3.2a - 3.2d), one can see an increase in bandwidth as the power is increased, as expected from above discussion. From these figures it is clear that the phase distortions originating from propagating through the PCF are mostly quadratic.

Note that the counter-intuitive broadening of the pulse as the spectrum broadens is due to the large quadratic phase profile from propagating through the PCF. If one were to assume a flat phase for all four cases, the the pulse length will decrease as expected.

3.2 Multiphoton intrapulse interference phase scan

Spectral phase of an ultrashort pulse can be determined by employing a multitude of pulse characterisation techniques. Of these techniques a Multiphoton Intrapulse Interference Phase Scan (MIIPS for short) is particularly suitable to reconstruct quadratic phase, as will be demonstrated in the following sections. This technique is a non-interferometric single beam method that utilises a pulse shaper (SLM in combination with a 4f-shaper), combined with second harmonic generation in an iterative manner to characterise and compress ultrashort pulses [17]. This single beam feature makes it attractive to employ in relatively compact systems.

For ease of reading spectral phase will be referred to only as phase, where as other cases will be referred explicitly in the appropriate context.

3.2.1 Access to spectral information

Performing a MIIPS requires a method of accessing the spectral content of the pulses, as well as modifying the phase of said content.

Access to the spectral components is gained by implementing a 4f-shaper in the optical setup. A 4f-shaper consists of grating and lens pairs arranged as illustrated in figure 3.3. The 4f-shaper, in short, allows the assigning of spatial coordinates to individual spectral components.

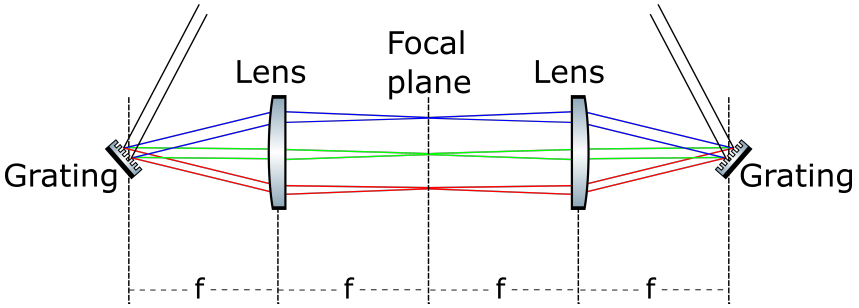


Figure 3.3: Schematic representation of 4f-shaper setup where components are specifically arranged in 4f positions allowing for access to spectral content of incoming pulse. Incoming pulses will exit the shaper unaffected if no phase object is present in the focal plane.

Adding a spatial light modulator (SLM) into the focal plane of the 4f-shaper allows for at will modification of the spectral intensity and phase of each component. SLM's are programmed with transfer functions to affect the chosen changes to the spectral components. These functions precisely define the chosen changes that the SLM has to apply to each spectral component.

The MIIPS technique was developed to make use of this combination to determine the phase of a pulse.

Both the 4f-shaper and SLM will be discussed in more detail in Chapter 4.

3.2.2 Phase reconstruction

The working principle behind this technique, MIIPS, is the discernible influence that a change in the phase will have on the second harmonic generated spectrum of the pulse. It is sensible then to write the expression connecting the phase to the SHG spectrum, $S^2(\omega)$, generated with a beta barium borate (BBO) crystal. The spectrum will be generated at $2(\omega_0 + \omega)$, but for convenience we center it around 0 [18]:

$$S^{(2)}(\omega) = \left| \int_{-\infty}^{\infty} |E(\omega + \Omega)||E(\omega - \Omega)| \times \exp i[\varphi(\omega + \Omega) + \varphi(\omega - \Omega)] d\Omega \right|^2 . \tag{3.1}$$

In words this expression adds up contributions of different frequencies ($\pm\Omega$) around a frequency ω to get the spectrum at 2ω, by looking at the amplidute,$|E(\omega \pm \Omega)|$, and the phase $\varphi(\omega \pm \Omega)$ at these frequencies. This SHG spectrum, at $2(\omega_0 + \omega)$, shows the connection between the total phase and the second-harmonic spectrum. Where the phase is zero, the SHG signal will be maximised. It is this property that MIIPS will exploit to reconstruct the phase of pulses to be characterised.

To simplify this expression a Taylor expansion of the phase is performed in order to approximate a simpler expression for the SHG signal:

$$\varphi(\omega \pm \Omega) = \varphi(\omega) \pm \varphi'(\omega)\Omega + \frac{\varphi''(\omega)}{2}\Omega^2 \pm ... \tag{3.2}$$

Substituting this into equation(3.1) produces a simpler relationship between the SHG signal and the total phase:

$$S^{(2)}(\omega) \approx \left| e^{2i\varphi(\omega)} \int_{-\infty}^{\infty} |E(\omega+\Omega)||E(\omega-\Omega)| \times \exp\left[i\varphi''(\omega)\Omega^2\right] d\Omega \right|^2. \tag{3.3}$$

Here even terms have cancelled out, higher order terms have been neglected and the first phase term is a constant factored out in front and represents a temporal shift and have no effect on the spectrum. This leaves a SHG signal related to the GDD of the pulse. From this approximation it is clear that MIIPS can accurately reconstruct quadratic phase distortions, making it ideal for the output from the ANDi PCF.

With the SHG signal now only related to the GDD, the condition for the phase retrieval is constructed as:

$$GDD = \varphi''(\omega) = 0. \tag{3.4}$$

This condition is exploited by adding a periodic reference phase function to the phase of the input or pump pulse, $\phi(\omega)$:

$$f(\omega) = \alpha \cos\left(\gamma(\omega - \omega_0) - \delta\right). \tag{3.5}$$

Parameters α and γ are the amplitude and scaling factor of the reference phase function and have chosen values relating to the pulse to be characterised. This sinusoidal reference function is suitable as it can be easily scaled and integrated. The phase retrieval condition then becomes:

$$GDD = \varphi''(\omega) = \phi''(\omega) + f''(\omega) = 0. \tag{3.6}$$

Introducing this reference phase function to the input pulse, cancels and reduces phase distortions in the output pulse. A maximum in the SHG signal is observed where the phase has been compensated for. Recording the SHG spectra of output pulses for different values of δ and combining these into a spectrogram constitutes a MIIPS trace, which is used to retrieve an estimation of the phase.

With $f''(\omega) = -\alpha\gamma^2 \cos\left[\gamma(\omega - \omega_0) - \delta(\omega)\right]$ already known, one can clearly see that:

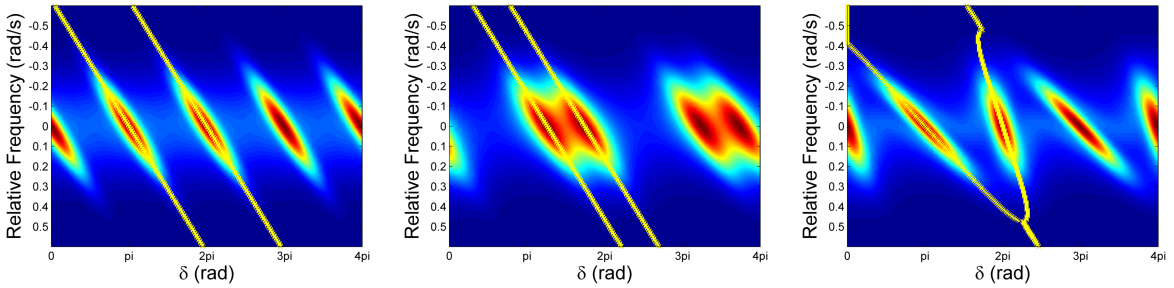
$$\phi''(\omega) = -f''(\omega) = \alpha\gamma^2 \cos\left[\gamma(\omega - \omega_0) - \delta_{max}(\omega)\right]. \tag{3.7}$$

This formula can be used to determine the GDD and in turn the phase for values of ω, by extracting from the MIIPS trace values for δ corresponding with ω where there is a maximum in $S^{(2)}$. Note that δ in equation(3.7) is now dependant on ω.

The extent to which MIIPS is able to theoretically resolve the GDD is dependant on the parameters of the reference phase function, α and γ. The maximum GDD that can be compensated for by MIIPS is:

$$|\varphi''(\omega)| = \alpha\gamma^2. \tag{3.8}$$

The best values to choose for γ is close to or equal to the expected pulse length.



(a) Transform limited pulse (b) Pulse with quadratic phase (c) Pulse with cubic phase

Figure 3.4: Figure of simulated MIIPS traces for arbitrary pulse with flat, quadratic and cubic spectral phase profiles respectively. Yellow crosses indicate maxima in trace for each spectral component, $(\omega - \omega_0)$, to determine corresponding δ.

Substituting retrieved values into $f''(\omega)$ for each ω produces the first estimate of the GDD. These values are illustrated in figure(3.4). Integrating the GDD twice will produce the first estimated spectral phase of the pulse, the negative of the phase to be exact, which can then be applied to the pulse with the same shaper, used to retrieve the phase, compressing the pulse. The retrieved phase however, is only an estimation of the phase, as higher order terms in $\varphi(\omega)$ are not taken into account. To compensate for this one can repeat the MIIPS procedure until the phase converges. One of the advantages of using this procedure is that it compresses the pulse while it characterises it.

From equation(3.7) one can see that for $\phi(\omega) = 0$, $\delta(\omega) = \omega\gamma \pm \frac{\pi}{2}$, which is why on MIIPS traces transform limited (TL) pulses are characterised by straight lines spaced π apart. This property is illustrated in figure(3.4a). For higher order phase this will no longer be true, where quadratic spectral phase shifts the spacing between structures in the MIIPS trace (figure(3.4b)) and cubic phase affects the slope of the line characterising the pulses (figure(3.4c)).

3.2.3 Simulations

Simulations of the MIIPS procedure are performed by mimicking the way a scan would be performed in the laboratory environment. An ideal shaper is considered, meaning that shaper effects are not taken into account and had 'infinite' resolution.

Simulated temporal profiles (as seen in figure (3.2)) are transformed to frequency domain by a Fast Fourier transform. In this domain we can easily manipulate the pulses

by simply multiplying the frequency spectrum with some transfer function. In this case the transfer function is the sinusoidal reference phase function ($f(\omega)$) used to perform a MIIPS. Multiplying the spectrum by the phase function, taking the inverse Fourier transform and squaring the result mimics the behaviour of the BBO crystal and produce the envelope of the resultant electric field. Taking the Fourier transform and squaring the results again transforms the result back to the frequency domain and produce the SHG intensity spectrum of interest. This process is then repeated by varying $f(\omega)$ for different values of δ ranging from 0 to 4π in 256 steps. A spectrogram is constructed by collecting the SHG spectrum for different values of δ. A peak finding algorithm is used to find maxima in the spectrogram as illustrated figure(3.4). As discussed previously, substituting these values into $f''(\omega)$ represents the first estimate of the GDD. After doubly integrating the GDD a first estimate of the phase is extracted. A new transfer function is then constructed by subtracting this reconstructed phase from $f(\omega)$ and multiplying this with the spectrum. This process is repeated until the residual phase is approximately zero.

We calculate the average residual phase by defining a region where the intensity is above a percentage of the peak intensity and computing the average phase in this region. These values are plotted per iteration as visual representation of the convergence.

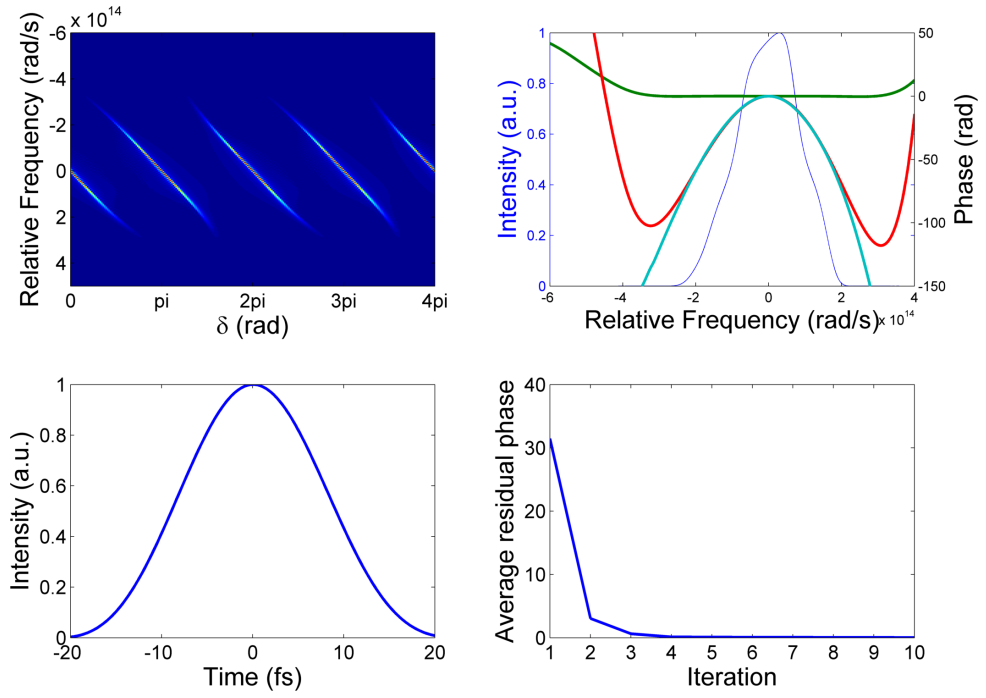


Figure 3.5: Simulated MIIPS measurement on PCF output from pumping at 800 nm with 50 mW power. Parameters for reference phase functions are $\alpha = 25$ rads and $\gamma = 15$ fs Top left shows MIIPS trace after 10 iterations while top right figure shows the spectrum with residual as well as total phase retrieved after 10 iterations. Bottom left image shows temporal pulse after phase correction with total phase retrieved. Bottom right image shows the average residual phase retrieved from MIIPS trace for 10 iterations.

It can be seen here that after 4 iterations the average retrieved phase from the MIIPS

trace is zero. With the knowledge of what the effect of different phase effects have on the MIIPS trace, it is immediately clear that the majority of the phase distortions have been corrected for. On the MIIPS trace one sees that the structures are, where the majority of the intensity lies, equidistant parallel straight lines. Further investigation shows that the retrieved phase (red curve) matches closely to the input phase(teal curve).

This means that the majority of the phase over the spectrum after 10 iterations is flat or approximately zero. This should produce a near transform limited temporal pulse. When comparing the temporal profile after compression with that of the simulated pulse before characterisation (figure (3.2e)) it is clear that the pulse has been compressed. The FWHM has reduce from approx. 800 fs to approx 14 fs - an order of magnitude smaller. The MIIPS measurement simulations are repeated for the same PCF outputs shown earlier.

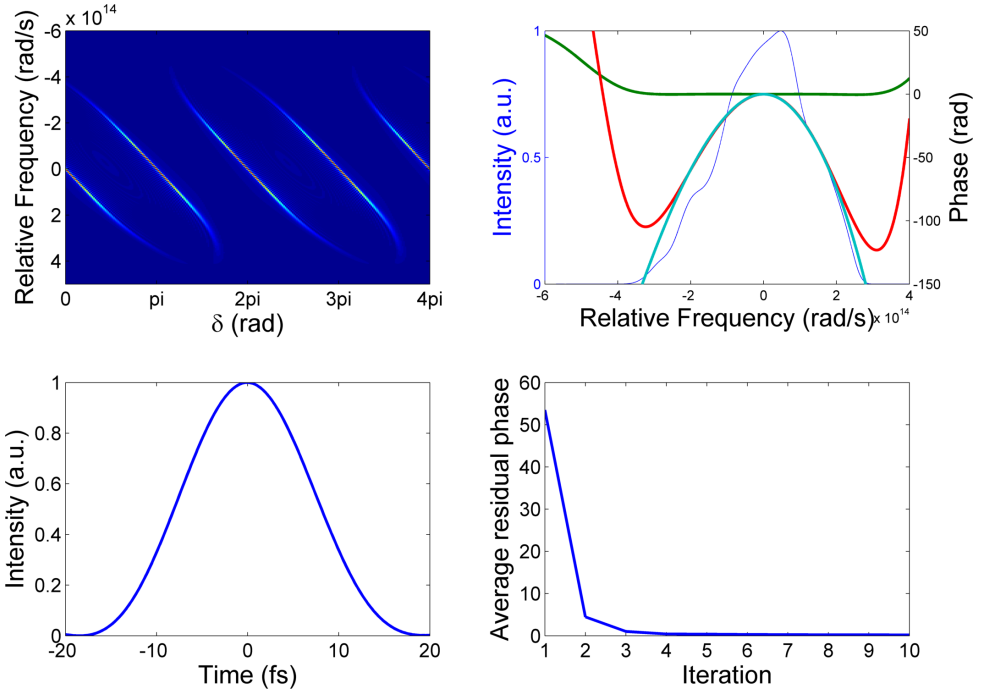


Figure 3.6: Simulated MIIPS measurement on PCF output from pumping at 800 nm with 100 mW power. Parameters for reference phase functions are $\alpha = 25$ rads and $\gamma = 15$ fs Top left shows MIIPS trace after 10 iterations while top right figure shows the spectrum with residual as well as total phase retrieved after 10 iterations. Bottom left image shows temporal pulse after phase correction with total phase retrieved. Bottom right image shows the average residual phase retrieved from MIIPS trace for 10 iterations.

For an increase in input power from 50 mW to 100 mW we again see that after 4 iterations the average residual phase converges to 0 average residual phase (figure (3.6)). This implies that the reference phase function parameters are still appropriate for this bandwidth and phase distortions. As for the temporal pulse we don't see an obvious decrease in pulse length and are still around 14 fs pulse length.

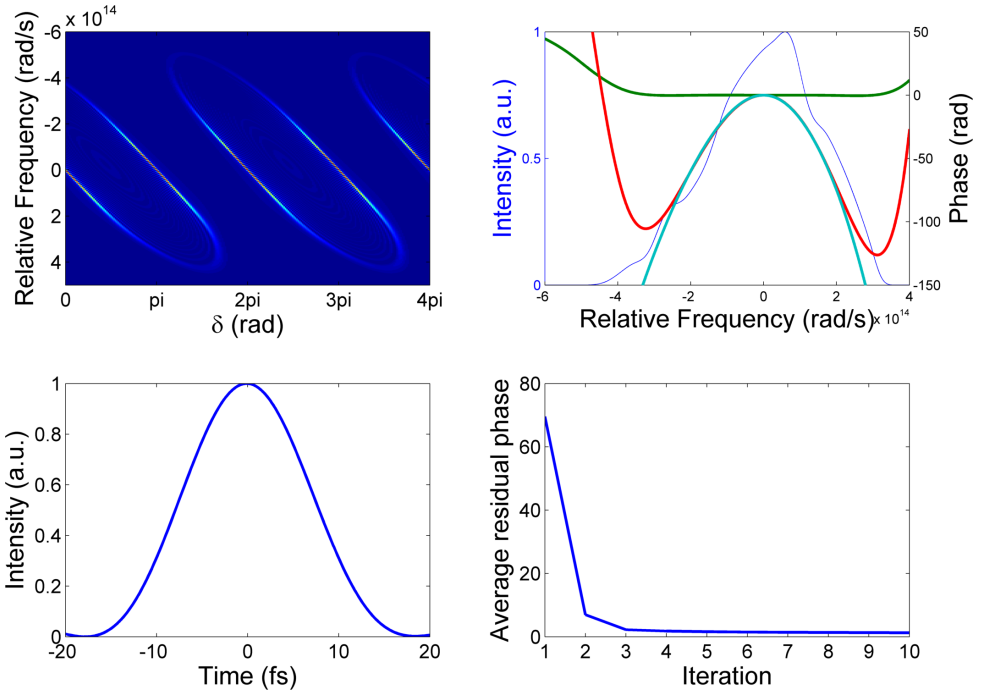


Figure 3.7: Simulated MIIPS measurement on PCF output from pumping at 800 nm with 150 mW power. Parameters for reference phase functions are $\alpha = 25$ rads and $\gamma = 15$ fs Top left shows MIIPS trace after 10 iterations while top right figure shows the spectrum with residual as well as total phase retrieved after 10 iterations. Bottom left image shows temporal pulse after phase correction with total phase retrieved. Bottom right image shows the average residual phase retrieved from MIIPS trace for 10 iterations.

Figure (3.7) indicates that increasing the bandwidth further, requires an adjustment in the parameters chosen for the MIIPS measurement. Specifically considering the phase curves we see that the retrieved phase (red curve) starts to deviate from the true phase (teal curve) on the edges where the intesity is still about a third of the peak intensity in the spectrum. This deviation is to be expected since the maximum retrievable GDD is dependant on α and γ according to equation (3.8).

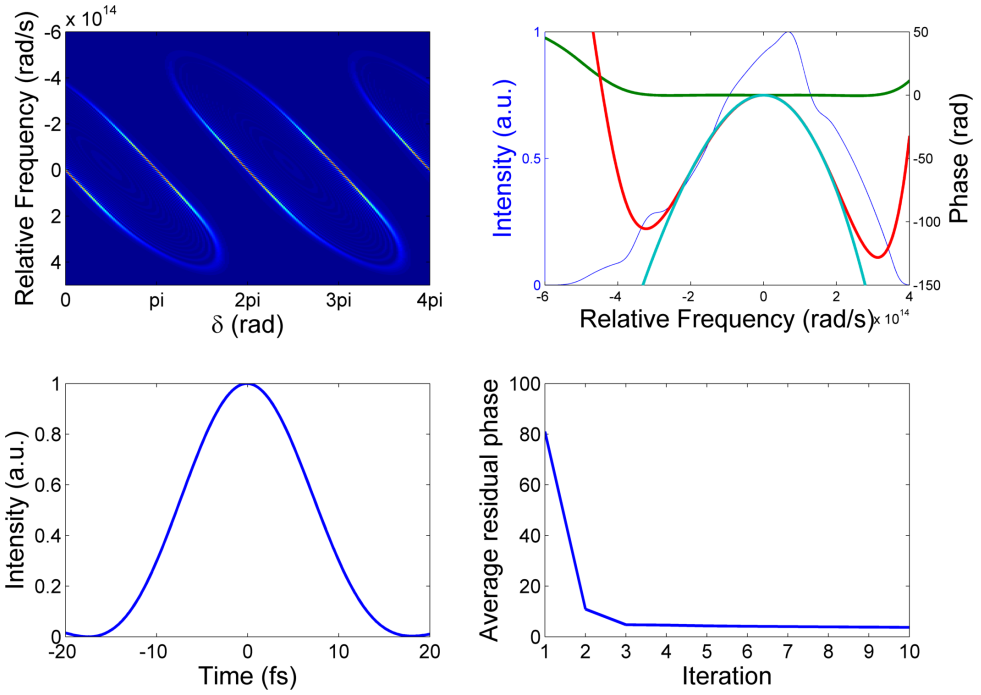


Figure 3.8: Simulated MIIPS measurement on PCF output from pumping at 800 nm with 200 mW power. Parameters for reference phase functions are $\alpha = 25$ rads and $\gamma = 15$ fs Top left shows MIIPS trace after 10 iterations while top right figure shows the spectrum with residual as well as total phase retrieved after 10 iterations. Bottom left image shows temporal pulse after phase correction with total phase retrieved. Bottom right image shows the average residual phase retrieved from MIIPS trace for 10 iterations.

Again for figure (3.8) we see that same phenomenon: For the same phase function parameters α and γ, the total reconstructed phase deviates from the real phase (teal curve) where the spectral intensity is still about a third of the peak intensity.

From these simulations we know that we should be able to adequately compress supercontinuum pulses with an average power of 100 mW with the phase function parameters $\alpha = 25$ rads and $\gamma = 15$ fs. This assertion can then be tested experimentally.

3.3 Gated MIIPS

3.3.1 Background

Under the approximation to second order phase in the SHG signal MIIPS only considers quadratic phase, limiting the effectiveness of retrieving higher order phase distortions. This and other limitations can be addressed by introducing a variation to the technique.

Gated-MIIPS (GMIIPS) is a gated variation on the MIIPS technique, where amplitude modulation - additional to the already added sinusoidal phase modulation - is applied to the spectrum to gate the spectrum around a specific frequency. This improves the polynomial expansion of the phase, enabling the use of higher values for γ and lower values for α minimising wrapping effects in the phase [18].

The transfer function applied to the pulse shaper has a Gaussian amplitude function added to the sinusoidal phase function and looks as follows:

$$M(\omega) = \exp\left[-\left(\frac{\gamma(\omega-\omega_0)-\delta}{\sigma}\right)^2 + i\alpha\cos\left(\gamma(\omega-\omega_0)-\delta\right)\right]. \tag{3.9}$$

This is introduced to equation (3.1) and produce the following expression:

$$\begin{aligned} G^{(2)}(2\omega) = \Big|\int_{-\infty}^{\infty} |E(\omega+\Omega)||E(\omega-\Omega)| \cdot \exp\left(-\frac{2\gamma^2\Omega^2}{\sigma^2}\right)\cdot \\ \exp i[\varphi(\omega+\Omega)+\varphi(\omega-\Omega)+2\alpha\cos(\gamma\Omega)\sin(\gamma\omega-\delta)]d\Omega\Big|^2. \end{aligned} \tag{3.10}$$

For this technique one also recovers a spectrogram for each iteration of the measurement. GMIIPS traces are characterised by one oval shaped structure, centred around $\delta = 0$ radians as can be seen in figure (3.10).

For the GMIIPS SHG the signal is maximised by the same condition as for MIIPS - maximum signal will be observed when the phase is zero or flat. This makes the implementation of GMIIPS straight forward if a MIIPS setup is already available. As in the retrieval of phase in the case of MIIPS, the δ values are collected for each ω, introduced back into $f''(\omega)$ and integrated twice to produce an estimation of the phase. This process will have an increased validity of second order expansion of the spectral phase.

In this procedure a gating parameter, σ, is introduced. This value is chosen relative to the sinusoidal reference function as half the period of the reference function. Sigma is then chosen in the following region:

$$0 < \sigma \leq \frac{\pi}{2}. \tag{3.11}$$

If one increases σ outside of this region, then one will eventually perform a MIIPS again.

This technique has computational advantages as the peak finding scheme is much simpler to that of the normal MIIPS.

3.3.2 Simulations

The simulation for GMIIPS is identical to that of the normal MIIPS procedure, except for the added amplitude function and the peak searching algorithm. Finding maxima on the GMIIPS trace is done by simply collecting all the maxima for each spectral component. This is possible to do considering that only one structure stretching the spectrum is to be expected.

To understand how GMIIPS works, consider the following figure.

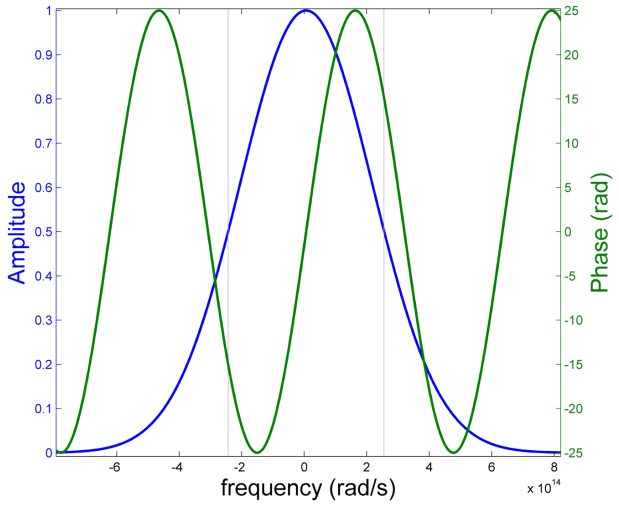


Figure 3.9: Amplitude (blue) and phase (green) functions applied to spectrum for GMIIPS. Both functions simultaneously scan over the spectrum

Here is illustrated the reference phase function as well as the amplitude function that is scanned over the spectrum. During the scan, the relationship between the two functions stays fixed. The phase function produces a normal MIIPS, while the Gaussian amplitude function acts as a type of spectral amplitude filter as the two scans over the spectrum.

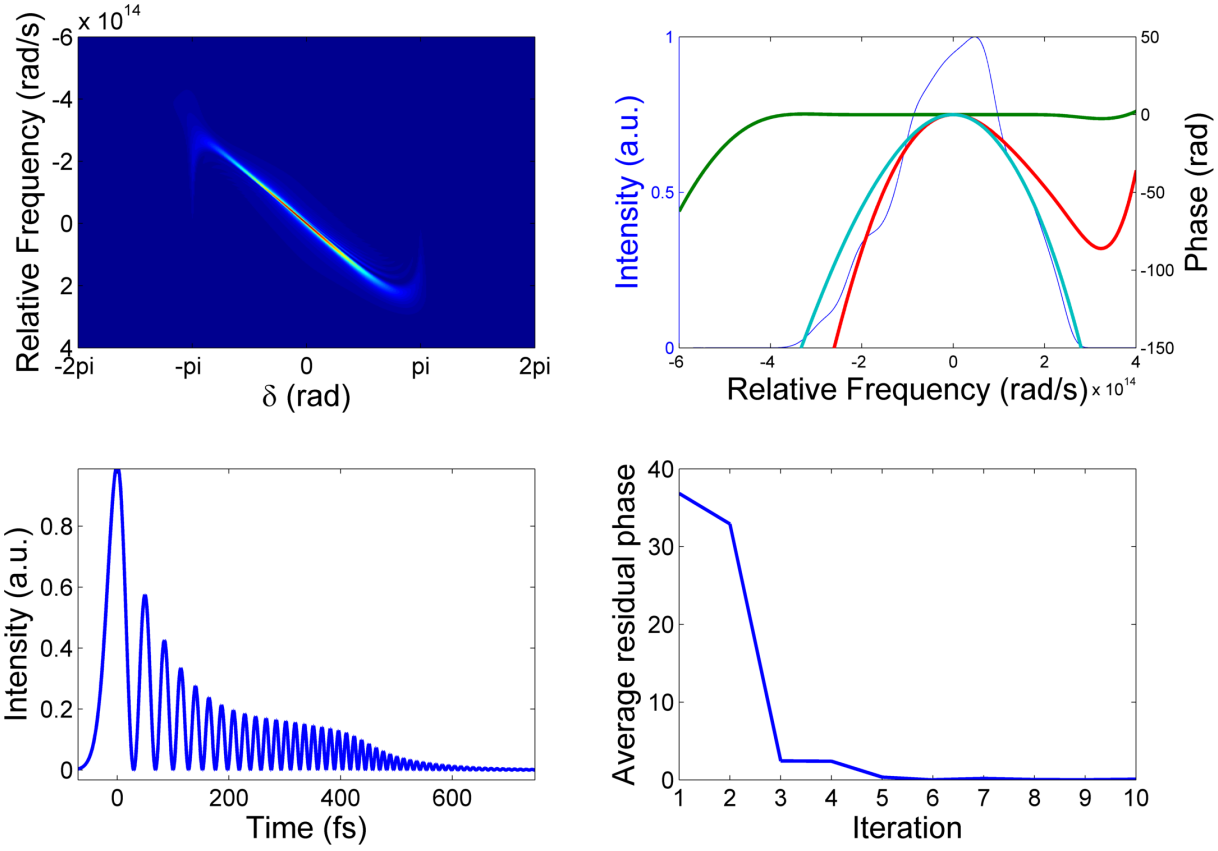


Figure 3.10: Simulation of GMIIPS with $\alpha = 25$, $\gamma = 10$, and $\sigma = 3$. In top right figure is the spectrum with total phase, residual phase and true phase as green, red and teal respectively, superimposed.

Simulations of GMIIPS for spectral output produced by 100 mW input power, suggest that there is no improvement of reconstruction as compared to that of MIIPS. The reconstruction is in fact worse. This can bee seen when considering the phase plot in figure (3.10). The total reconstructed phase, red curve, is visibly different from the true phase of the pulse, teal curve. With the residual phase, green curve, being flat implies that there is no more information to be extracted from the GMIIPS trace.

The parameters for this simulation were optimised to reconstruct the phase as close as possible. One can however use the simulations to see which values for σ work the best and it would seem that for the best reconstruction of the phase, one must choose σ in such a way to include more than half the period (figure (3.9)).

3.4 Single scan MIIPS

3.4.1 Phase retrieval

Ideally one would like to perform one scan of the pulse in order to retrieve the spectral phase. This is advantageous as it reduces the time it takes to perform time consuming experiments. It would also be able to retrieve the phase of pulses with phase distortions which pulse shapers cannot compensate for and thus not be able to retrieve. Such a retrieval technique has recently been developed and is similar to the FROG reconstruction algorithm [19]. This method aims to produce an identical MIIPS trace to that of the experimentally constructed when retrieved phase is substituted into equation (3.1). For this procedure only one experimental MIIPS trace (M_{exp}) and the spectral amplitude of the pulse is required.

The procedure is paraphrased from Comin et al. [19]:

1. Extract first estimate of GDD, $\varphi_0^{''}(\omega)$, from M_{exp}.
2. Integrate $\varphi_0^{''}(\omega)$ twice to produce first estimate of spectral phase, $\varphi_0(\omega)$.
3. Use measured spectral amplitude in combination with $\varphi_0(\omega)$ to generate a simulated MIIPS trace, M_{n-1}^{sim}.
4. GDD extracted from M_n^{sim} will be used as a feedback term, $\varphi_n^{''feedback}(\omega)$, used to update the extracted GDD.
5. Comparing this to the the extracted GDD from M_{exp} allows for an estimation of an error, $\Delta\varphi_{n-1}^{''}(\omega)$
$$\Delta\varphi_{n-1}^{''}(\omega) = \varphi_{n-1}^{''feedback}(\omega) - \varphi_0^{''}(\omega) \tag{3.12}$$
6. A new GDD can then be guessed by updating the previous GDD with the error term.
$$\varphi_n^{''}(\omega) = \varphi_0^{''} - k \cdot \Delta\varphi_{n-1}^{''}\omega \tag{3.13}$$

Where k is a numerical constant used to optimize the convergence of the algorithm.

The above steps are repeated until error term is smaller than a chosen tolerance. We know the minimum variation of the GDD extracted using MIIPS is set by

$$min[\Delta\phi''(\omega)] = -\frac{\gamma^2 \cdot \alpha}{N_\delta} \tag{3.14}$$

The tolerance is thus set to this value meaning convergence will be reached when

$$|\Delta\phi_0^{err}(\omega)| \leq \frac{\gamma^2 \cdot \alpha}{N_\delta} \tag{3.15}$$

3.4.2 Computational phase reconstruction

Simulating the SSMIIPS procedure is also similar to that of the MIIPS procedure. A normal MIIPS is performed, with the code from the MIIPS simulation, to generate a MIIPS trace. This trace is then fed through the process described above to generate a new trace to compare with. This is done iteratively until no more information can be extracted from the traces.

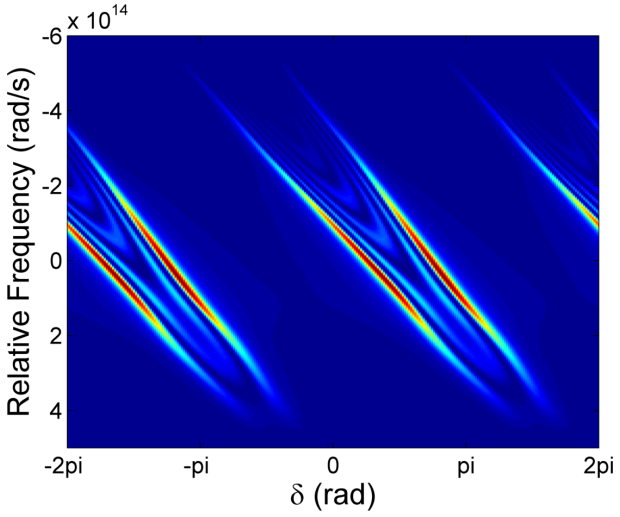


Figure 3.11: Simulated MIIPS trace generated using PCF output corresponding to 100 mW input power and reference phase function parameters $\alpha = 25$ and $\gamma = 15$.

The following simulation shows the results obtained after 10 iterations of the retrieval procedure. A representation of the pulse in time is illustrated by subtracting the reconstructed phase from that of the spectral phase. Average residual phase is again calculated by taking an average of the residual phase, in this case the difference between the true phase and the reconstructed phase, for some region of interest.

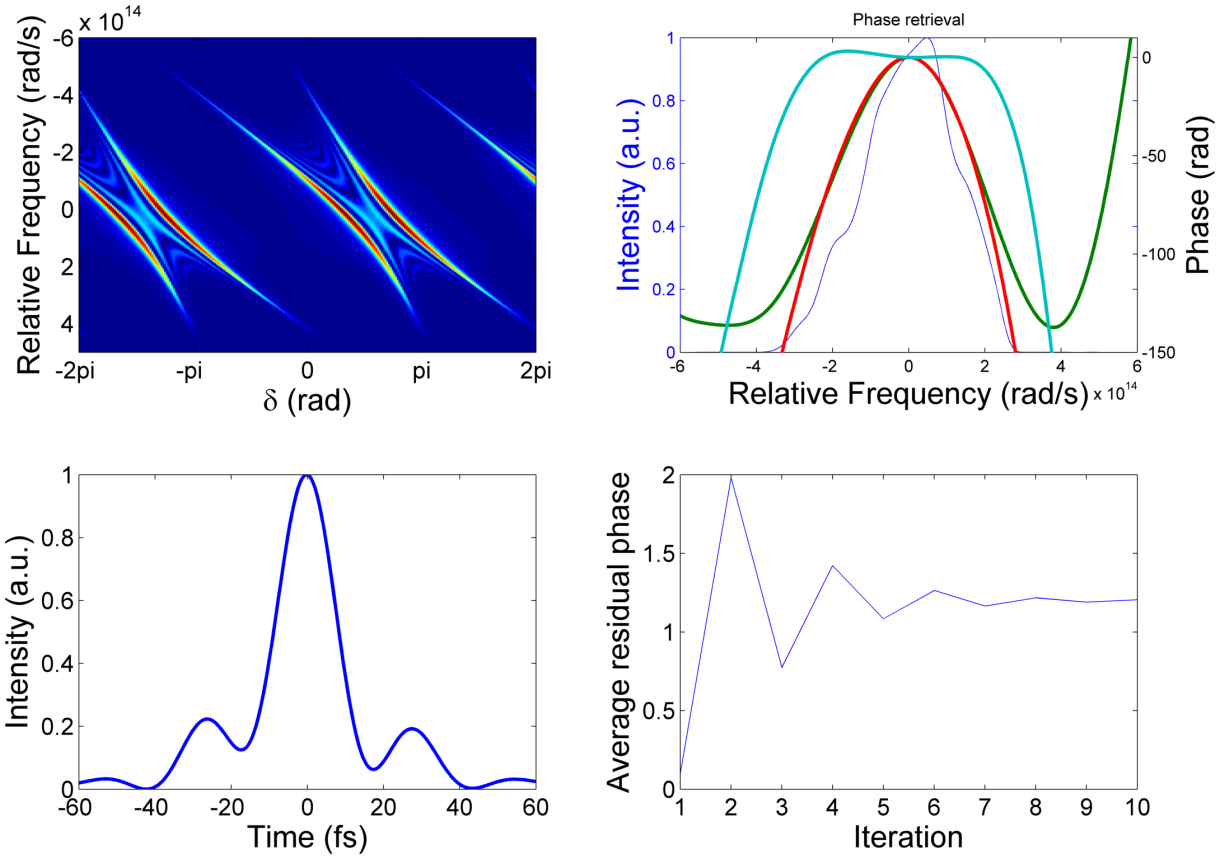


Figure 3.12: Simulated single scan MIIPS procedure performed on PCF output corresponding to 100 mW input power, with $\alpha = 25$, $\gamma = 15$, and $k = 0.5$. Reconstructions converges after about 10 iterations. Superimposed over the spectrum is the reconstructed phase, red, the actual phase, green, and the difference between the two, teal.

This method of reconstructing the phase seems to be successful - a lot more so than the GMIIPS procedure.

3.5 Interferometric frequency-resolved optical gating

To verify the retrieved phase produced by performing a MIIPS another pulse characterisation technique can be used to test it. A robust technique to do so is an interferometric frequency-resolved optical gating (IFROG) measurement. Performing an IFROG over a FROG (of which IFROG is a variant), will become clear when considering the experimental setup. IFROG works on the basis of spectrally resolving a collinear interferometric autocorrelation [20], from which conventional FROG information can be extracted and iteratively retrieve the electric field of the pulse.

Consider the following pulse (with spatial information omitted):

$$E(t) = \tilde{E}_o(t) \exp(i\omega_o t). \tag{3.16}$$

One can express the IFROG signals as:

$$I_{IFROG}(\omega, \tau) = \left| \int_{-\infty}^{\infty} (E(t) + E(t-\tau))^2 \exp(-i\omega t) dt \right|^2. \tag{3.17}$$

Taking the Fourier transform of equation(3.17) and expanding the result produce:

$$\begin{aligned} I_{IFROG}(\omega,\tau) &= 2|E_{SH}(\Delta\omega)|^2 + 4|E_{FROG}(\Delta\omega,\tau)|^2 \\ &+ 8\cos\left[(\omega_0 + \frac{\Delta\omega}{2})\tau\right] Re[E_{FROG}(\Delta\omega,\tau)E^*_{SH}(\Delta\omega)\exp\left(i\frac{\Delta\omega}{2}\tau\right)] \\ &+ 2\cos\left[(2\omega_0 + \Delta\omega)\tau\right]|E_{SH}(\Delta\omega)|^2. \end{aligned} \tag{3.18}$$

where $\Delta\omega = \omega - 2\omega_0$. Note that the different components have already been labelled. In doing a Fourier transform on the IFROG trace we aim to isolate the SH-FROG component from the data. In equation(3.18) the first two components are the delay independent background and the SH-FROG trace. The third and fourth terms are interferometric contributions and represent the side-bands in the Fourier transform of the IFROG trace.

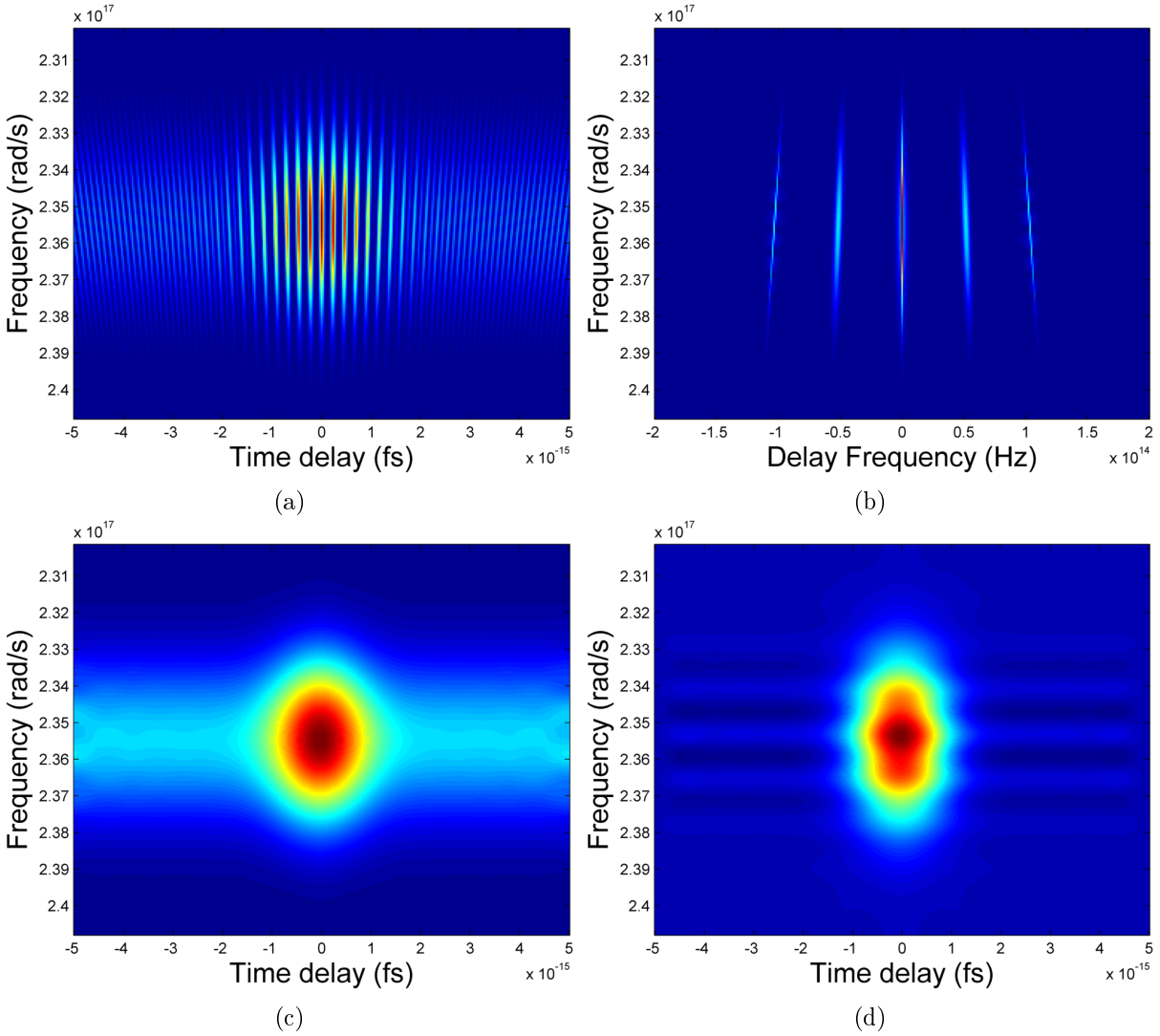


Figure 3.13: Figure of (a)simulated IFROG trace. (b) Fourier transform of (a) for each spectral component, with dc baseband consisting of FROG components as well as SH background. (c) IFROG trace with interference removed. (d) Extracted FROG trace.

From equation(3.18) one can see that a traditional SH-FROG trace is contained within the IFROG trace. To retrieve the pulse information from an IFROG trace one needs to extract a FROG trace first. To do so we take a Fourier transform of the IFROG trace along time delay axis (that is a Fourier transform for each spectral component). One can see from figure(3.13b) that the interference terms in the Fourier domain are sufficiently separated from the dc band. This means that these terms can simply be discarded. Once this has been done a Fourier transform back into the time domain (time delay) produces a spectrogram such as figure(3.13c) which contains E_{SH} as well as E_{FROG}. Simply subtracting the background at large delay times from the trace produces a FROG trace. From this FROG trace one can then extract the electric field to compare with MIIPS results.

In this chapter we discussed the theoretical background regarding the generation, characterisation and compression of coherent supercontinuum light. Simulated PCF output was used to test the efficacy of MIIPS to characterise and compress supercontinuum pulses produced by ANDi PCFs. This serves as a proof of principle before applying this technique in the lab environment. Variations of the MIIPS technique was put throught the same computational testing and seem to have no immediate benefit over using MIIPS.

Th IFROG technique, which is particularly suitable in the planned setup, will be used as an extra measure to validate the degree of compression of MIIPS in the lab environment.

Chapter 4

Experimental setup

With the theory of this project now well established through literature study and simulations, this chapter will discuss how supercontinuum generation, characterisation and compression is realised in the laboratory environment.

We start our discussion with the pulse source, a Spectra-Physics femtosecond Tsunami laser, of which we are interested to know what the output characteristics are. This will be determined through a simple measurement of the spectrum as well as a measurement of the pulse duration (at FWHM) through means of an IAC (discussed in chapter 2). Continuing on, the specifications of the PCF is presented accompanied by the experimental methodology to determine the characteristics of interest. This is followed by the remainder of the pulse compressor components that consists of a 4f-shaper in combination with a one dimensional spatial light modulator (1D SLM).

4.1 Tsunami Laser

The Spectra-Physics Tsunami laser is a titanium-doped sapphire solid state laser pumped by a 532 nm cw Millennia Vs laser.

The pump laser, with Nd:YVO_4 as gain medium, has an output of 1064 nm which is converted through SHG in a temperature-tuned lithium triborate (LBO) nonlinear crystal. This second harmonic is generated through intracavity frequency doubling, achieving high conversion efficiencies[21].

This output at 532 nm is coupled into the Tsunami laser cavity and is used to pump the birefringent Ti:Sapphire rod. This laser is actively mode locked by use of an acousto-optic modulator (AOM), allowing the laser to operate for long periods of time with an 80 MHz pulse repetition rate. Present inside the laser cavity is a prism sequence and variable slit. This allows for tuning of the laser to different wavelengths[2]. For the experiments that follow, the laser has been tuned to a centre wavelength at 800 nm.

Tsunami specifications	
Average Power	1 W
Peak Power	> 125 kW
Pulse Width	< 80 fs
Tuning Range	735 – 840 nm
Repetition Rate	80 MHz
Spatial Mode	TEM_{00}

Figure 4.1: Laser specifications for chosen operating mode[2]

Measurements throughout the study were taken after a few hours of laser operation to ensure temperature stability and thus a stable output. The spectrum of the Tsunami tends to drift over time. This drift is minimised by having the laser reach this temperature stability. Special care is taken to have the spectral output as broad as possible and centred around 800 nm, by tuning the slit and prism alignment in the laser resonator. This in turn ensures that the pulse duration is optimised to its shortest length.

4.2 Interferometric Autocorrelator (IAC)

The interferometric autocorrelator is setup to match the schematic representation in figure 4.2 - resembling a rapid-scan Michelson interferometer[10]. It is vital that the mirrors M1 and M2 are the same distance away from the beam splitter (BS). This ensures that the two copies of the input beam have the same optical path length (when the speaker/driver is at rest) meaning that the two copies will after reflection be spatially and temporally overlapped in the beamsplitter and propagate together to the two photon detector. To measure an IAC signal with the oscilloscope, the signal generator drives the speaker with a sinusoidal wave of amplitude 4V and frequency 10Hz - shifting the variable pulse forwards and backwards around the pulse to be measured.

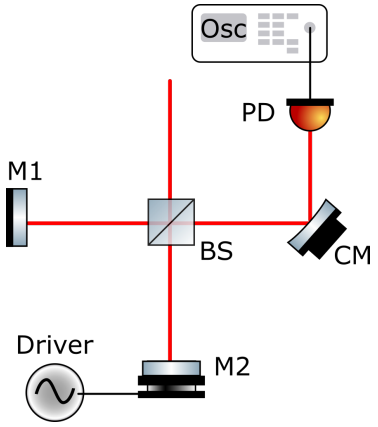


Figure 4.2: Schematic representation of an interferometric autocorrelator consisting of mirrors (M1 static and M2 mounted on a speaker); a curved mirror (CM); a beam splitter (BS) 45° to incoming beam; a signal generator (driver); a two photon detector (PD) and an oscilloscope (Osc).

To ensure that IAC measurements are as accurate as possible, a few procedures can be followed. An obvious example would be to ensure that the beam propagates parallel to the optical table, by use of two pinholes placed some distance apart before coupling into the autocorrelator. One can iteratively adjust the alignment of the system until a peak to background ratio of 8:1 is reached or as close as one is able to achieve this. An effective way to see whether the system is aligned, is to monitor multiple signals on the oscilloscope. If the distance between these signals alternate, then one must adjust the overlap to address this issue. Once the alignment is optimally adjusted, then one can extract the IAC signal from the oscilloscope for determining of the temporal pulse duration.

4.3 Pulse compressor

With the limitation on the pulse width set by the bandwidth output from the laser, novel methods have to be employed in order to create pulses shorter than this limit. This limitation is determined by the frequency bandwidth of the output as discussed earlier. The pulse compressor has been designed to circumvent this limit by employing a photonic crystal fibre in the setup. In short the pulse compressor works as follows: 'Transform limited' pulses[1], produced by the laser, propagate through the fibre. The PCF increases the bandwidth of the pulses, depending on the input power, and also stretch these pulses out in time. These dispersed pulses then propagate through a 4f shaper-SLM setup that iteratively determines and compensates for the phase distortions of the pulse - creating pulses shorter than those produced by the laser.

The components are discussed in more detail separately.

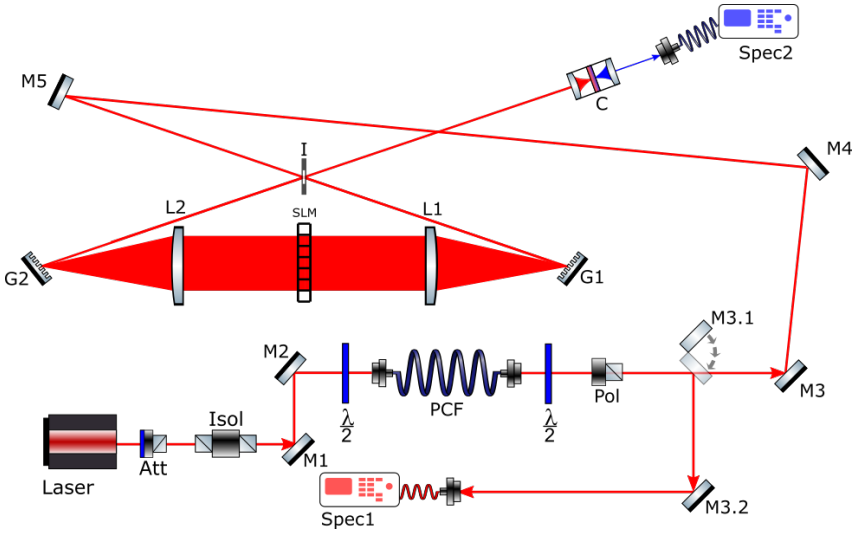
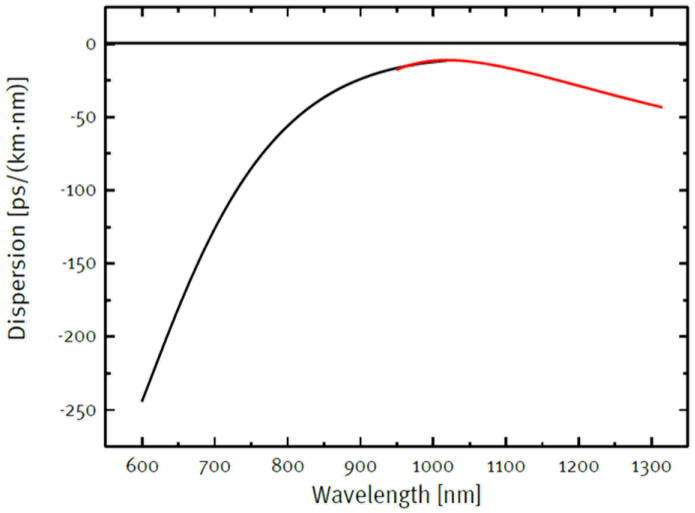


Figure 4.3: Schematic representation of pulse compressor showing the most important components of the setup.

[1]These pulses are approximately transform limited.

4.3.1 Photonic Crystal Fibre (PCF)

The first essential part of the pulse compressor is the photonic crystal fibre. The photonic crystal fibre used in this setup is a 15 cm long all-normal dispersion (ANDi) fibre with a flat dispersion around 1050 nm.



(a) Figure of tailored flat dispersion for a pump wavelength of 1050 nm.

Specifications	
Material	Pure Silica
Core diameter	2.3 ± 0.3 µm
Flat normal dispersion	1050 nm

(b) Table of PCF specifications

Figure 4.4: Figure containing PCF specifications used in this study[1].

Taking into account that the pump source for the PCF, produced by the Tsunami laser, is centred at 800 nm, one expects the supercontinuum pulses produced by the PCF to be stretched out in time. This stretch is caused by spectral phase distortions, i.e. the spectral components are no longer in phase. One can conclude this by considering the dispersion profile (in figure(4.4a)). This PCF has a tailored flat phase around 1050 nm and has a sharp change in gradient towards lower wavelengths. This further illustrates the need for the pulses to be characterised and compressed as the pulses produced by the PCF are no longer transform limited.

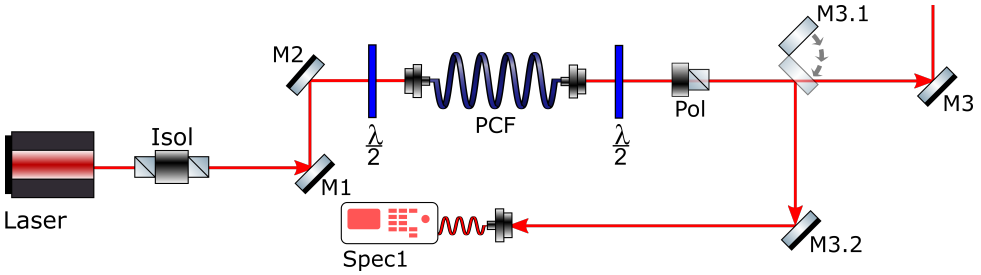


Figure 4.5: The PCF setup consists of a Laser; an isolator (Isol); mirrors (M1 to M3); two halfwave plates ($\frac{\lambda}{2}$); a polariser (Pol) and a spectrometer (Spec1). These components can stand alone in order to study the PCF.

In experimental realisation of the setup, figure (4.5), we see that the PCF has a halfwave plate on the incoming and outgoing ends. This is due to the PCF not being a polarisation maintaining fibre. This means that the incoming linear polarisation will not be maintained throughout the propagation. We condsider the PCF to be a birefringent material that is twisted along its length and postulate that the light couples into a fast and slow axis in the birefringent material, producing two polarisation components that are temporally decoupled. One can see this effect when analysing the polarisation of the outgoing pulses, by using the halfwave plate and polariser.

Changing the input polarisation and/or power of the pulses focused into the fibre, affects the output of the fibre. In order to use the fibre effectively one must study the effect this has on the output of the fibre. Characterisation of the PCF can be done by flipping mirror M3.1 into the beam path, redirecting the light to the spectrometer (Spec1) for spectral analysis of the fibre output.

A power meter can easily be placed after the polariser in order to study the polarisation characteristics. This is done by setting the incoming halfwave plate to a polarisation and measuring the power as a function of the polarisation of the outgoing halfwave plate over 90 degrees. As stated earlier, we postulate that the exiting beam exists of two separate beams with a 90 degree shift in polarisation with respect to each other. Analysing this should produce a superposition of two $\cos^2$ graphs. This process is repeated iteratively for the incoming halfwave position over 90 degrees. We assume that the halfwave plates are perpendicular to the beam, implying that setting the halfwave plate to 91 degrees is equivalent to setting it to 1 degrees etc.

Analysing the spectrum of the output as a function of input power is done by simply changing the attenuator (Att) to a desired setting, recording the power and then the spectrum with the spectrometer. One can do this for all possible input powers and input polarisations. This is however not necessary as we are interested in the spectrum for the input polarisation which produce the highest power output in a single polarisation axis.

When aligning the PCF one has to take special care when initially coupling the light into the fibre. Working at low power reduces the risk of damaging the face of the PCF. A damaged face may reduce the coupling efficiency. It is also important to note that the

spatial profile of the output should be uniform and without spatial distortions. If these are present, it may indicate that the face of the PCF has to be cleaned. If the cleaning procedure does not work, then one has to resort to a fibre polishing procedure similar to one described by Thorlabs [22].

4.3.2 4f Shaper

The 4f shaper grants access to essentially each frequency component of the laser pulses. It does this by separating the pulse into its frequency components by shining the beam onto a 600/mm diffraction grating with 750 nm blaze. A cylindrical lens placed a focal length, f, away from the grating performs an optical Fourier transform mapping the angular dispersed frequency components to specific spatial coordinates in the focal plane. Placing another identical lens a distance f away from the focal plane focuses the light to a point onto another grating combines the spectral components back into a pulse.

Determining what the spatial coordinate of each spectral component in the focal plane is, opens up the possibility to change the spectral phase of the pulse and affect some change in the temporal domain.

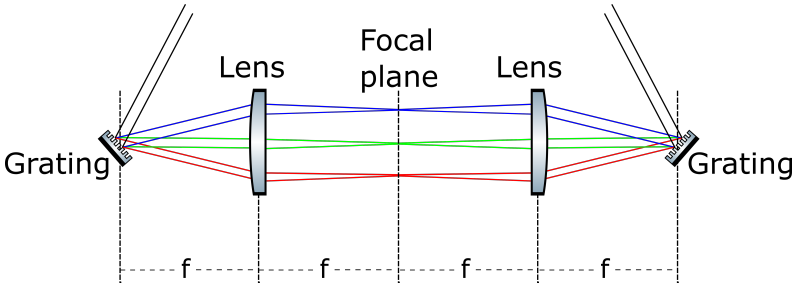


Figure 4.6: 4f Shaper represented illustrates ability to separate light into its spectral components and mapping each component to a spatial coordinate in the focal plane.

Placing these components a distance f apart, gives rise to the name of the shaper, but is essential for it to function properly. With this configuration one is assured that the wave front in the focal plane is planar and not curved (i.e. each component focuses in this plane); it also assures that if no phase distortion in the focal plane is introduced, that the pulse will exit unchanged after propagating through the system. Changing the distance of one of these components has the effect of creating a spatial spread of spectral components over the exiting beam. This effect can be seen while aligning the shaper and is an effective way of testing whether the system is properly aligned. One can use an objective to magnify the spatial profile and monitor what happens if an object moves through the focal plane. If you are able to see this movement on the magnified beam, then the shaper is ill aligned. Ideally one would want only the intensity to decrease, implying the spectral components overlap spatially.

In order to be able to dynamically change the phase relationship of the spectrum, a 1D SLM is introduced in to the focal plane. With this one is able to program phase functions onto the SLM which will translate into a physical change on the temporal pulses.

4.3.3 Spatial Light Modulator

The spatial light modulator is a computer programmable device which allows for phase and/or amplitude modulation. The SLM used in the pulse compressor is a 1D Jenoptic SLM-S with a resolution of 640 strips. This device is based on the control of two birefringent nematic liquid crystal (LC) layers, allowing for simultaneous phase and amplitude modulation.

A strip is essentially a pixel on the SLM display and is an array of nematic liquid crystals that are squeezed together by two electrodes. These crystals are aligned ± 45 degrees in the x-y plane. It is important thus to have the incoming beam horizontally polarised. Here the x plane refers to the plane parallel to that of the optical table and the y-axis perpendicular.

If a voltage is applied across these electrodes then the crystals will rotate towards the plane of the beam propagation, z [23]. This creates a difference in path length between the two components propagating through the crystals. If one introduces a linear polariser along the x-axis in the outgoing beam path, then one is able to perform phase and amplitude modulation independently - establishing the working principle behind the SLM.

In order to have an operational device one must have a calibration curve for both LC displays voltage vs. phase. In this application of the SLM one is also interested in a calibration of the wavelength position on the SLM. This is simply done by programming a slit on the SLM that scans over the pixel array while collecting the corresponding spectrum with a spectrometer.

4.3.4 SLM assisted IFROG

As briefly mentioned in chapter 3.5 a 1D SLM in a 4f-shaper combination is able to perform an IFROG measurement. This is possible due to the ability of the SLM to modulate the phase as well as the amplitude of the spectral content of a pulse.

In order to perform an IFROG, one has to apply an appropriate transfer function (TF) to the SLM. The desired result from such a function would be to emulate a double pulse with a variable delay between them. This can be achieved by using the following phase function:

$$M(\omega, \tau) = \frac{1}{2}(e^{i\omega\tau} + 1) \tag{4.1}$$

Applying this TF to the SLM will create two pulses with a temporal distance of τ between them. Scanning in an appropriate time interval around 0 and collecting the spectrum into a spectrogram for each time delay will constitute an IFROG measurement.

Chapter 5

Experimental Results

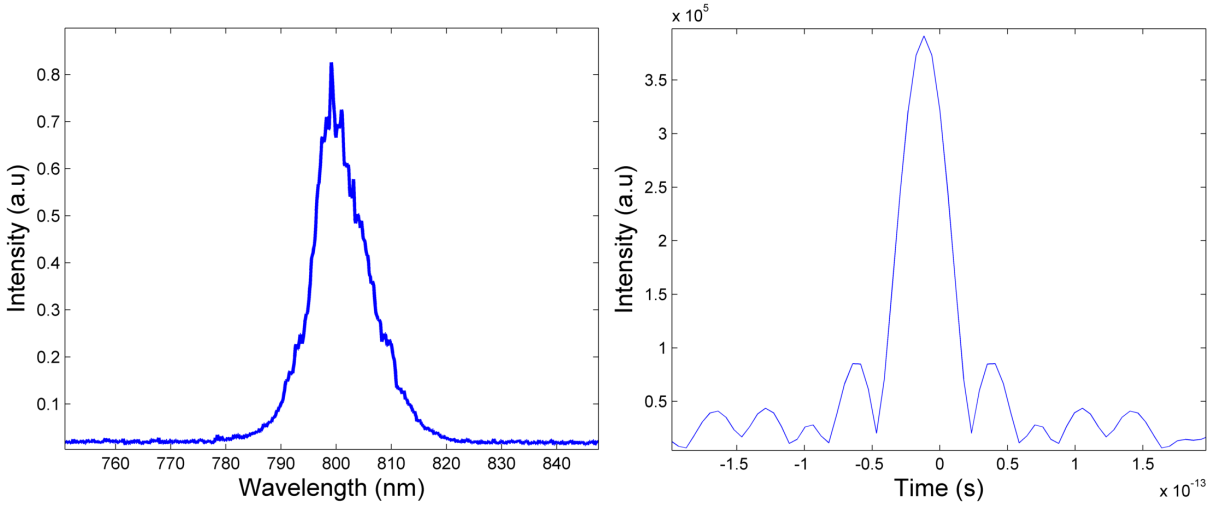
In this chapter the results obtained from characterising and testing the major components of the pulse compressor is presented.

A comprehensive study of the pulse compressor was conducted to determine how the adjustable parameters affect the effectiveness of the system. With this knowledge one is able to implement the pulse compressor in other systems that require ultra short pulses or the dynamic adjustability in pulse lengths and shapes that this compressor offers.

More specifically: the results obtained from studying the PCF is presented as well as the results obtained from testing the MIIPS procedure. An SLM assisted IFROG was performed on a compressed pulse, in order to determine by alternative methods whether the pulses have indeed been compressed.

5.1 Interferometric autocorrelation (IAC)

The spectral output of the Ti:Sapphire laser, as measured, is centred around 800 nm with a spectral bandwidth, $\Delta\lambda$, of about 12 nm. Performing a Fourier transform on the measured spectrum, figure (5.1a), and assuming a flat spectral phase in the process, produces a representation of what a transform limited pulse corresponding to this spectrum would look like (figure (5.1b)). This gives an indication of the shortest possible pulse with the corresponding spectrum.



(a) Measured spectrum of Ti:Sapphire laser centred around 800 nm.

(b) Transform limited pulse associated with measured spectrum with a pulse duration in the order of 80 fs.

Figure 5.1

Performing an autocorrelation measurement as discussed in chapter 2.6 provide the means with which the pulse duration can be determined. Measuring the intensity of the pulse overlapped with itself at different time delays will produce the result shown in figure (5.2). It is assumed that the pulses are Gaussian and the IAC data is fitted with a Gaussian envelope.

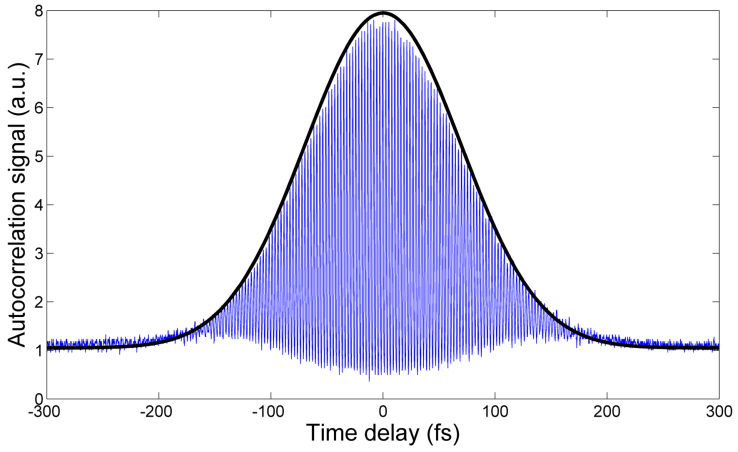


Figure 5.2: Measured autocorrelation signal (blue) with Gaussian envelope (black) fitted to data. IAC result has a background to peak ratio close to 1:8. Starting parameters for Gaussian envelope was chosen to match this ratio.

Results obtained from performing an IAC measurement show a background to peak ratio close to 1:8. This ratio is characteristic for the IAC and is a required feature in the experimental results to be considered accurate. For an IAC it its assumed that recombined beams spatially overlap perfectly after recombination in the beam splitter. Non-optimal spatial overlap of pulses can be the cause of the slight deviation from the 1:8

ratio. This means that results obtained from IAC's deviating from this ratio will produce an estimation of the pulse duration within some margin of error that correlated to how far it deviates.

The full width half maximum of the Gaussian fit, $\Delta\tau_{IAC}$, is extracted and converted to the FWHM of the envelop of the electric field, $\Delta\tau$, using the appropriate conversion factor for a Gaussian shaped pulse. The extracted pulse duration from this measurement is $\pm$ 81.98 fs.

5.2 PCF characterisation

In our experiment a non polarisation maintaining PCF, pumped by a fs laser is used to produce coherent supercontinuum pulses. In order to implement this fibre into the experimental setup, one has to know the characteristics of the output from the fibre. We are interested in knowing how the outgoing polarisation changes relative to the incoming polarisation. With this knowledge one can identify an optimal input polarisation. At this polarisation one can also study the spectrum of the fibre output as a function of input power. Ideally one would like to know a set of parameters that produce the highest power throughput for a polarisation component (or the highest extinction ratio), coinciding with a broad spectrum that has a spectral profile that is more or less uniform (does not dip to zero in the middle of the profile).

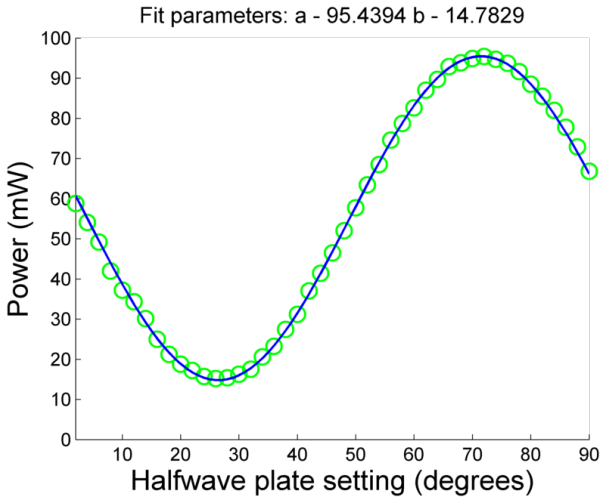
5.2.1 Polarisation characterisation

Determining the polarisation characteristics of the fibre output is executed by analysing the polarisation of the output for each polarisation input. This analyses is performed by setting the incoming halfwave plate to a polarisation and then measuring the output power for different output halfwave plate settings from 2 to 90 degrees. The polarisation in this case is not absolute, but relative so some axis on the half wave plate. This means that should any of the halfwave plates be replaced, then one has to shift the results to match the new point of reference. Repeating these measurements for different incoming halfwave plate settings provide a full picture as to the output polarisation characteristics for different input polarisations. Presented are two of these measurements in figure (5.3). These data sets are fitted with the following model - a superposition of Malus' law:

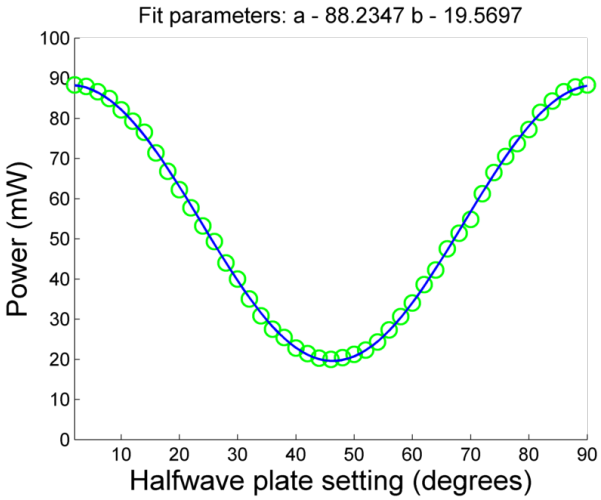
$$a \times \cos\left(x\frac{\pi}{90} + c\right)^2 + b \times \cos\left(x\frac{\pi}{90} + \frac{\pi}{2} + c\right)^2 \tag{5.1}$$

where a, b and c are fit parameters. This model is used in accordance with postulation discussed in Chapter 4.3.1, wherein the fibre output consists of two temporally decoupled orthogonal polarisation components.

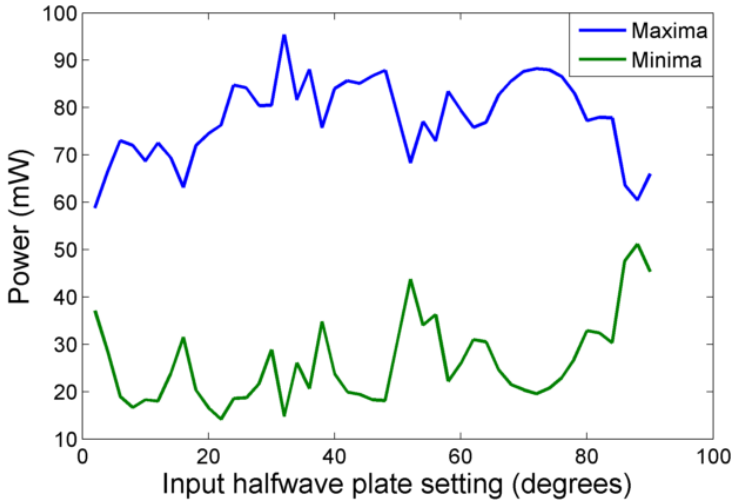
For each measurement the input power was kept at a constant of 200 mW with a coupling efficiency of 51%. This coupling efficiency is a fraction of the incoming power, just before the coupling lens, over the outgoing power, just after the beam collimator.



(a) Polarisation characteristics with input halfwave plate at 32 degrees. With a maximum power output of 95.43 mW.



(b) Polarisation characteristics with input halfwave plate at 72 degrees. With a maximum power output of 88.23 mW.



(c) Collected maxima and minima for input halfwave plate setting from 2 to 90 degrees.

Figure 5.3

At first glance of figure (5.3c) it would seem that the ideal incoming halfwave plate setting would be at 32 degrees, with a corresponding output halfwave plate setting at 73 degrees (figure(5.3a)), for maximum power throughput. Further investigation, however, shows that it is not in a region of stability. If one considers figure (5.3c) one can see that for incoming halfwave plate settings 65 to 80, the coupling of light into different propagation axes is more or less well behaved. It also turns out that at the maximum of this region, with an incoming setting at 72 degrees, is a spectrum output with an almost uniform intensity throughout the spectrum. With this it is implied that the spectrum

does not dip to zero anywhere in the profile. This property ensures a smoother temporal profile for the compressed pulse and makes it ideal for calibrating the wavelength position on the SLM.

5.2.2 Effects of input power

The next step in characterising the fibre output is to investigate the influence of the input power on the spectrum of the fibre output spectrum. From theory we expect the the spectrum to increase in bandwidth as the input power is increased.

Setting the halfwave plates to the determined setting of input and output halfwave plates to 72 and 46 degrees respectively, and capturing the spectrum as a function of input power produce the following result:

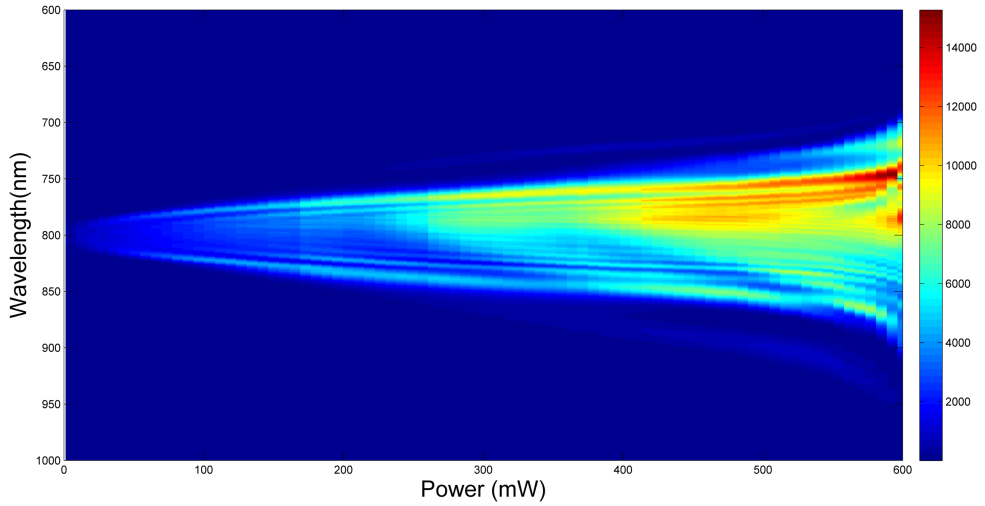


Figure 5.4: A collection of spectra has been combined into a spectrogram illustrating the output spectrum's dependence on input power. Colours represent the intensity where red is a high intensity and blue low intensity. Taking a vertical slice will produce the spectrum for the corresponding power.

The results, as presented in figure (5.4), clearly show that as the power increases, so does the bandwidth. It also illustrates the power of the PCF. With only 600 mW input power, which not the full output power of the laser, the PCF has increased the bandwidth from around 15 nm to roughly 200 nm. This opens up the possibility to dynamically control the pulse length and consequently the peak intensity, by choosing a bandwidth with corresponding expected pulse length.

5.3 MIIPS

As discussed previously, the chosen method of characterising and compressing the pulses is MIIPS. This procedure is implemented through a computer interface created in Labview which is able to control both the SLM and spectrometer, to take measurements of the spectrum as the phase function scans over the SLM as explained in section 3.2 of chapter 3.

5.3.1 Phase function parameters

It is of interest to know how the parameters chosen for the reference phase function in the MIIPS procedure influences the MIIPS trace and the speed of convergence. The ultimate goal is to have a functional user friendly pulse compressor that can be implemented in other systems. Knowledge of the these parameters will contribute to a fuller understanding of how to achieve this goal.

On an intuitive level we know that changing the amplitude of the reference phase function changes the width of the characteristic MIIPS structures within the MIIPS trace while changing γ changes the slope of these structures. For this investigation the input power was chosen as 100 mW. At this input power we are confident that the SLM is able to compensate for phase distortions present in the supercontinuum produced by the PCF. The output spectrum from the PCF with this input power is presented below(figure(5.5)).

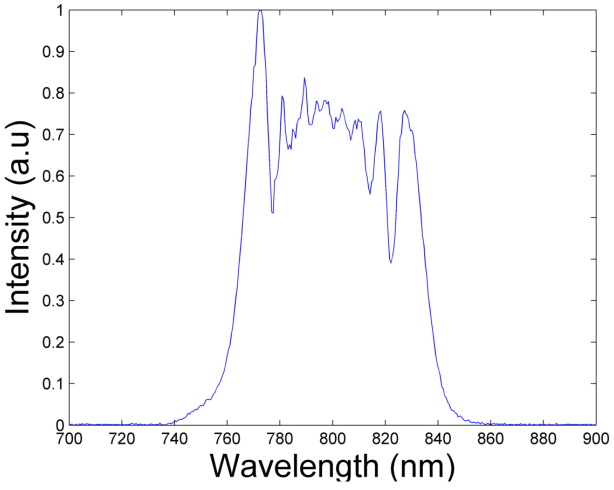


Figure 5.5: Spectrum of PCF output with input power at 100mW and 51% coupling efficiency. Halfwave plates set to 72 and 46 degrees for incoming and outgoing plates respectively.

A set of phase function parameters were tested by performing MIIPS for each of the chosen combinations. The input pulse parameters were kept constant during this investigation. This ensures that the measurements that follow have the same spectrum and enable us to make comparative conclusions from the results.

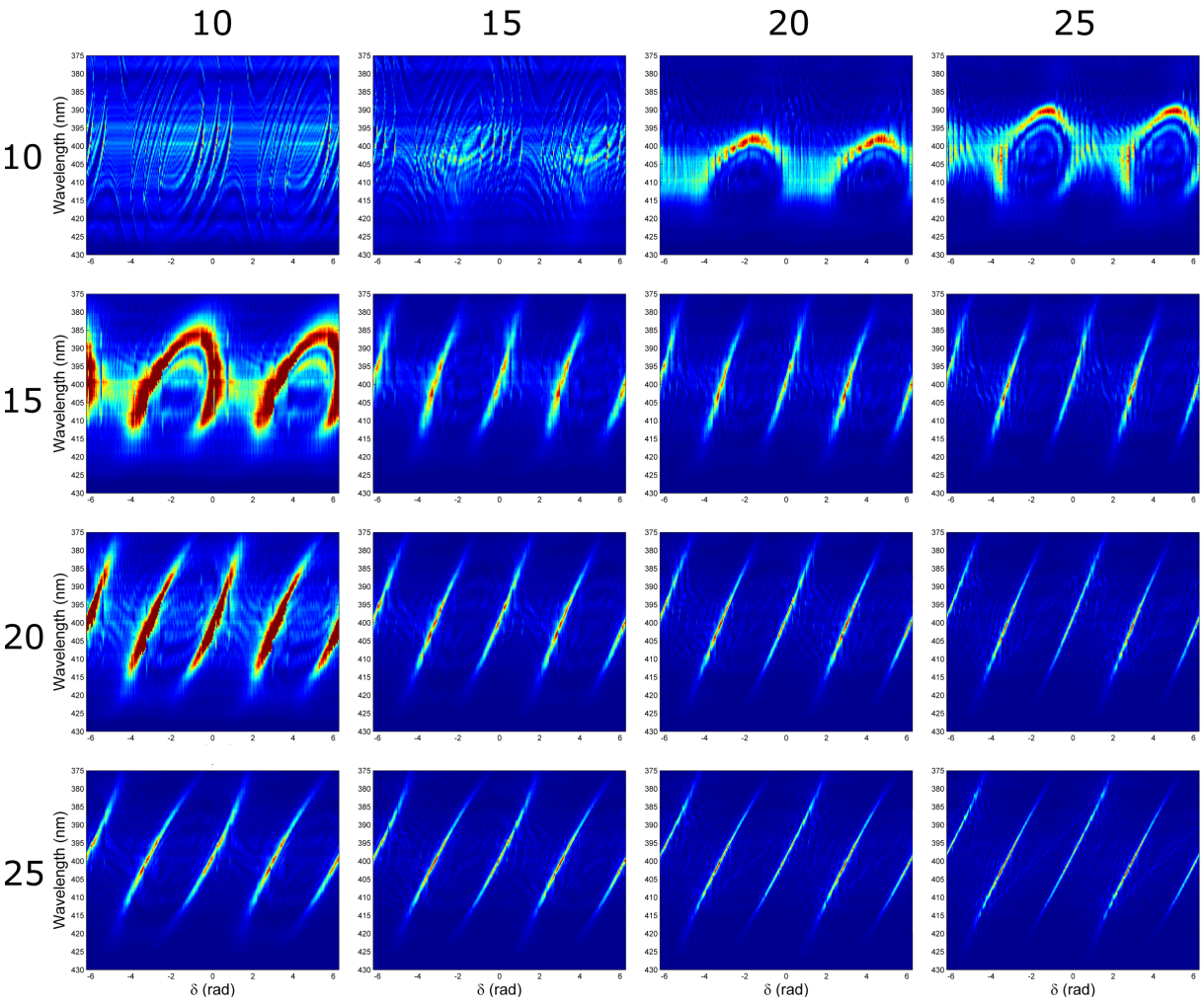


Figure 5.6: Presented a set of MIIPS traces after 10 iterations. Rows represent the scaling factor, γ, while the columns represent the phase function amplitude, α. For all traces the x-axis plots the scanning parameter, δ, of the reference phase function, while the y axis plots the wavelength, and colours from blue to red represent intensity from low to high.

Figure (5.6) graphically illustrates the play off between the two phase function parameters. For ease of referencing, specific traces in figure (5.6) will be referenced by using (Column/α, Row/γ) as shorthand notation.

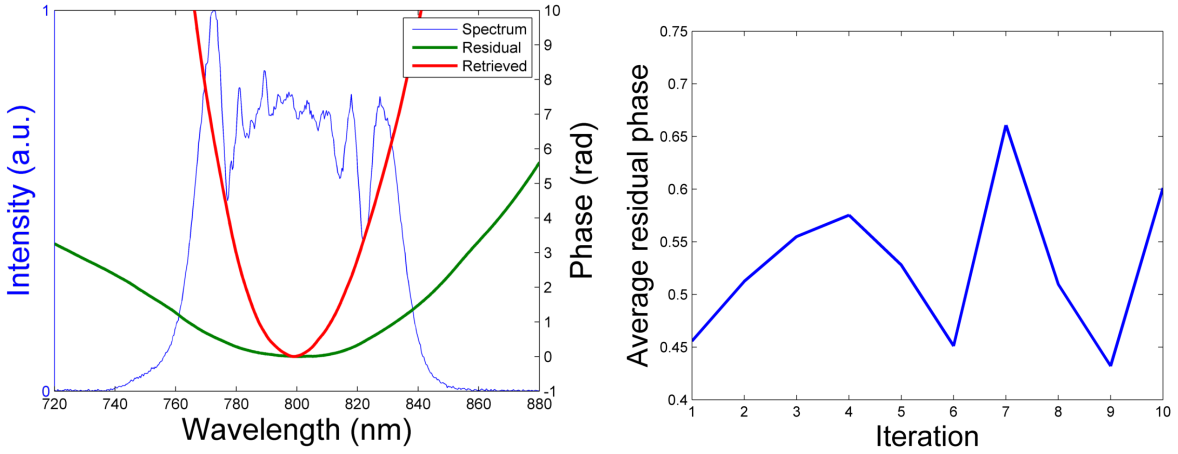
From these traces one can conclude that for small values of α (column 10) one can retrieve the GDD by sufficiently increasing γ.

For the traces with smaller γ we see that the characteristic MIIPS structures are not as well resolved, for the same amount of iterations, compared to other traces with higher γ values. Comparing this to traces with smaller α values, we see that traces with small α values are more effectively resolved with increased γ values, than those with smaller γ and increased α values. This is to be expected when considering equation (3.8), where the maximum GDD that one is able to retrieve is related to the square of the scaling factor, γ.

In trace (10,10) one can see the SLM wrapping effect, with structures being successively repeated three times. This wrapping effect is due to the limitation of the SLM. Values of

γ that are small will translate into a steep phase function to be programmed on the SLM. If this function causes a phase shift of more than 2π between consecutive SLM pixels then the phase will be wrapped and warp the MIIPS trace as seen here.

A closer look at the retrieved phase of such a measurement, provides one with the insight as to what we are able to retrieve from such a trace.



(a) Total retrieved (red) and residual (green) phase after 10 iterations of MIIPS procedure correlating with trace (10,10).

(b) Average residual phase per MIIPS iteration.

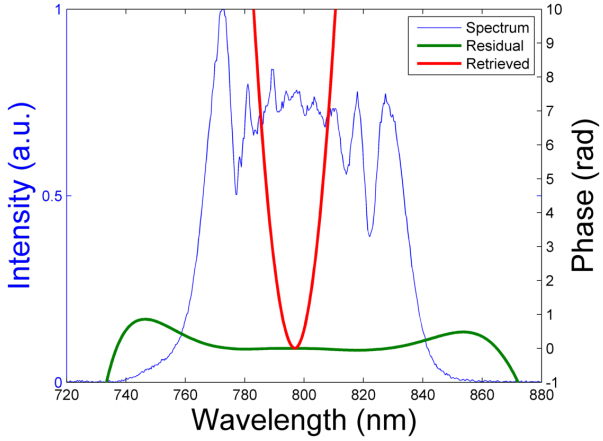
Figure 5.7

The average residual phase, as discussed in chapter (3), is calculated by computing the average residual phase where the spectral intensity is a percentage of the peak intensity.

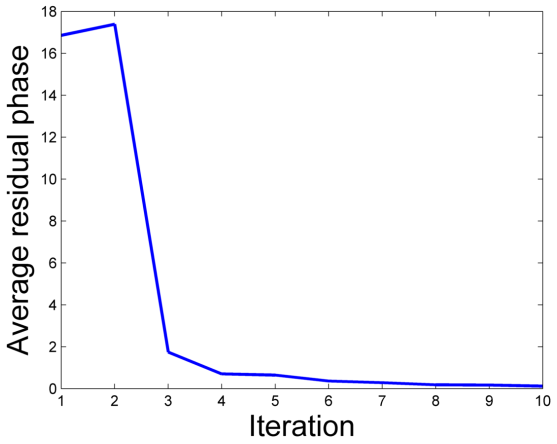
One is able to retrieve some of the phase of the pulse as can be seen here in figure (5.7a). The total phase is however not as steep as it should be as will be seen in figures (5.8a) and (5.9a). The reason one is able to retrieve phase from this measurement is the fact that a MIIPS like structure is present in the field of the trace where maxima are searched for. This structure is, however, not correct as can be seen from the phase that one can retrieve from this measurement. Essentially what is happening here, is that we are cumulatively adding phase per iteration of wrongly retrieved phases and appears to be similar to an accurate MIIPS measurement.

Using the traces in figure (5.6) one is able to identify which combinations of α and γ were successful in compressing the pulses. Take trace (25,20) for instance. In this trace the features are equidistant, more or less parallel throughout, and approximately straight lines. This indicates that the phase distortion were compensated for throughout approximately the entire spectrum. Compare this with an exaggerated trace (10,25). Here the structures are equidistant, but moving away from the central wavelength (up or down the y axis) one sees the structures bending towards each other indicating that only parts of the phase distortions of the spectrum surrounding the central wavelength has been compensated for. One can see this effect when analysing the retrieved phase for the different parameter combinations as well.

Looking at figure (5.8a) we see the above mentioned effect. The residual phase (green) in the centre of the spectrum is flat, while moving to the sides (which translates to up and down in the traces) we see that the phase curve curves up. This curvature in the phase is what causes the characteristic MIIPS structures in trace (10, 25) to curl towards each other. Another MIIPS iteration for (10,25) should produce a flatter residual phase and MIIPS structures that are straighter.



(a) Total retrieved (red) and residual (green) phase after 10 iterations of MIIPS procedure correlating with trace (10,25).

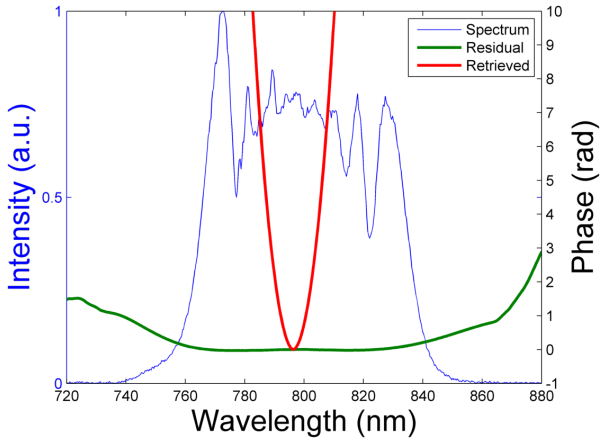


(b) Average residual phase per MIIPS iteration.

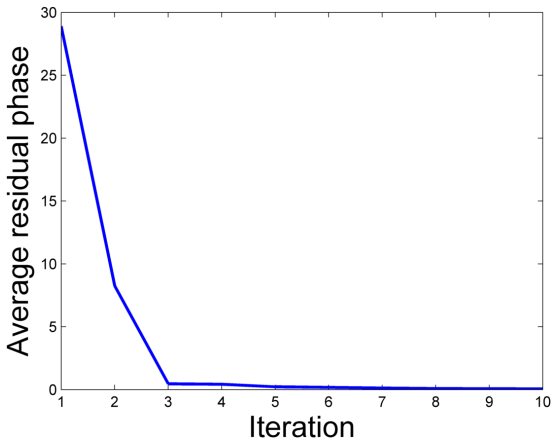
Figure 5.8

In figure (5.9a) we see that this effect is less pronounced with the phase being flatter over a larger portion of the spectrum. This translates to MIIPS structures that are straighter in (25, 20) than what we see in (10, 25).Comparing these two measurements we see that the chosen phase function parameters performs much better for (25,20) than for (10, 25). The average residual phase graphs illustrate this point much better.

The average residual phase for (25,20) was already close to zero after 3 iterations (figure(5.9b)), while the average residual phase for (10, 25) was only close to this number after 6 iterations (figure(5.8b)).



(a) Total retrieved (red) and residual (green) phase after 10 iterations of MIIPS procedure correlating with trace (25,20).



(b) Average residual phase per MIIPS iteration.

Figure 5.9

To gauge which parameters are best suited for this spectrum we can compare the average residual phase of the different cases at the third iteration. At this iteration we have already seen that (25,20) performs well and have already retrieved the majority of the phase. The reason for not choosing the 10^{th} iteration is that at this point the majority of the scans are already close to 0 average residual phase. At this point some scans can incorrectly appear to function better than others due to the average phase oscillating around some value. Looking at an earlier iteration gives you insight on the convergence.

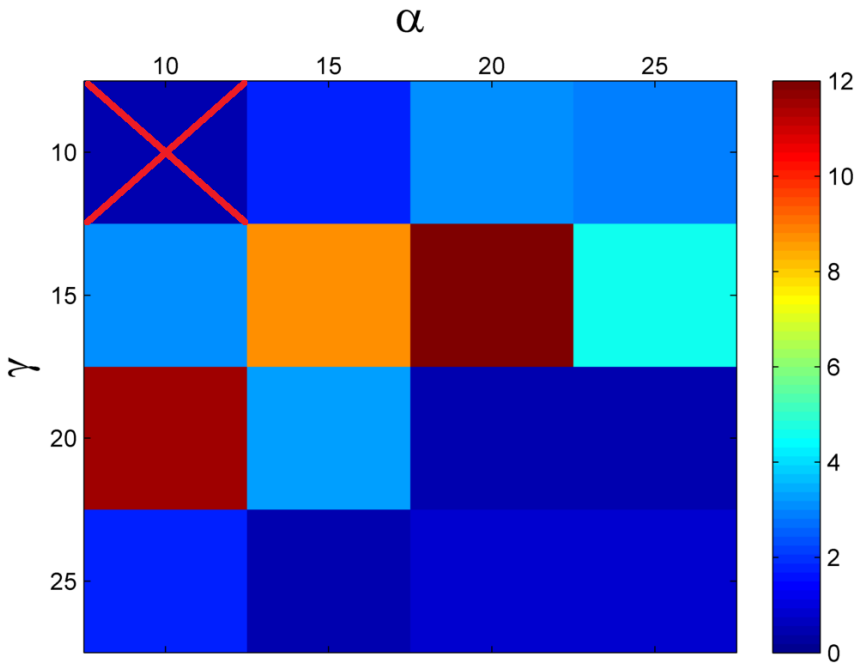


Figure 5.10: Illustrated here is the average residual phase of each parameter combination for the 3^{rd} iteration of the MIIPS. Colours correspond to phase value as indicated by the colour bar. Dark blue represents zero whereas dark red represents 12 radians.

We can neglect conclusions made from (10, 10) for previously discussed reasons. As for measurements (15,25) and (20,20), however, we see that these measurements are performing as well as (25, 20) at this iteration. This means that these three parameter sets can be used for implementation of the pulse compressor. A more concise way of deciding which of these parameters to use, is by comparing the average residual phase curves of these parameters, as the representation in figure (5.10) gives no information of what happens before and after this iteration.

5.3.2 MIIPS validity

We see from the theory and performing simulations that flat residual phase correlates with flat phase in the output pulses, which means that the pulses should be compressed. To test the validity of this claim one has to verify, by other means, whether the pulses have in fact been compressed. The most obvious would be to employ a well established pulse characterisation technique, such as a SPIDER[24] or FROG[8], to extract the phase and compare it with the retrieved phase from the MIIPS measurements. We have, however, opted to use other methods.

One such method is employed to indirectly test whether MIIPS accurately retrieves the supercontinuum pulse's phase. This method entails introducing a cube of well characterised glass with known dimensions into the beam path of an already compressed pulse. The glass cube will act as a phase element and chirp the pulse. The expected phase distortions from the glass cube can easily be calculated and compared with the MIIPS retrieved phase. With this we are able to gain insight as to the accuracy of the phase retrieval.

With the MIIPS simulation code at hand one can determine whether the method should in principle be able to retrieve the phase. This simulation is performed by using a measured spectrum similar to that used in the previous section, with the phase added manually. This phase corresponds to the expected phase distortion from the glass. Calculating the phase from the glass and adding it to the spectrum is done in the following way:

$$\sqrt{I(\omega)} \times e^{i(\omega^2 \times GDD)} \tag{5.2}$$

Where $I(\omega)$ is the intensity spectrum, ω is the frequency vector, and GDD is the GVD multiplied by the length of the cube (as discussed in Chapter 2).

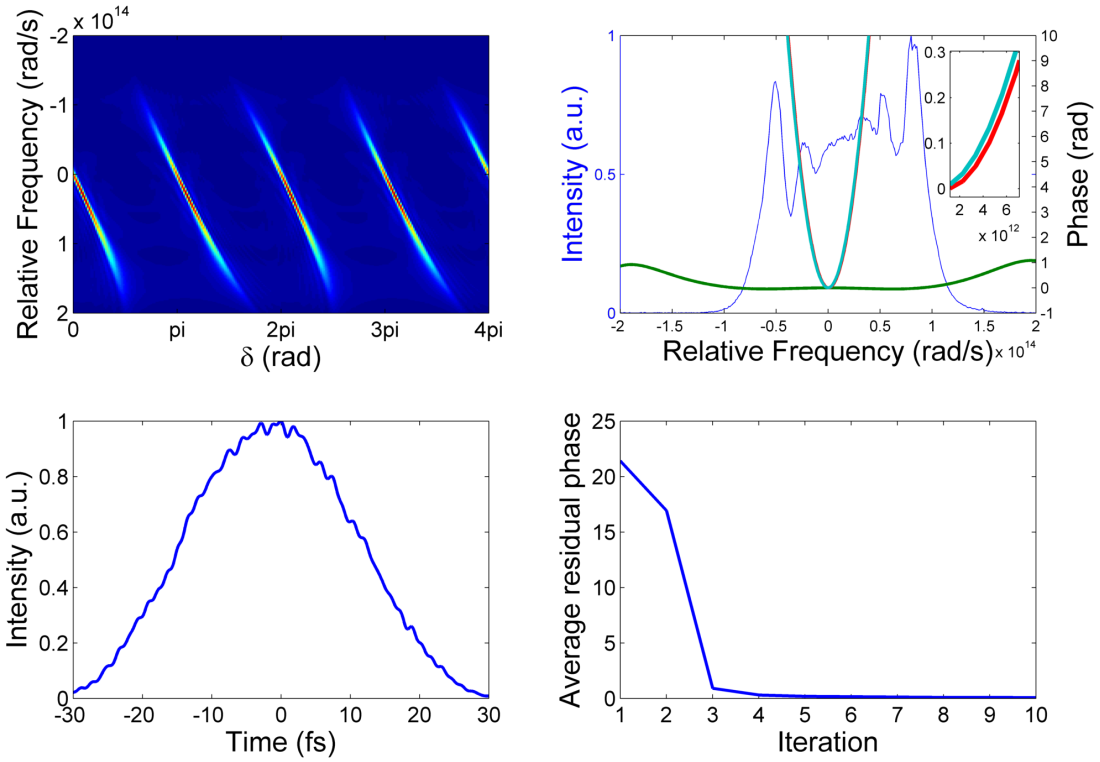


Figure 5.11: Simulation of 10 MIIPS iterations of transform limited pulses through 2 cm x 2 cm x 2 cm SF6 glass cube with a GVD of 327.68 fs^2/mm [3]. Here the red, green and teal phase curves represent the total retrieved, residual and calculated phase respectively.

From the phase curves in figure (5.11), one can see that the extracted phase and the glass phase are indistinguishable. A zoomed in section is plotted to show that both are in fact present. One can also see that the residual phase is flat and has converged to zero. This implies that MIIPS should be able to adequately retrieve the phase from the glass.

It is important when adding the glass cube to the beam path, that the beam is perpendicular to the glass surface. If the glass cube is not correctly placed into the system it can change the beam path and will require the system to be re-calibrated, as the light on the SLM will shift. This shift on the SLM means that the frequency components will fall on different SLM pixels. In addition, this shift will make it impossible to estimate the added phase rendering this method obsolete.

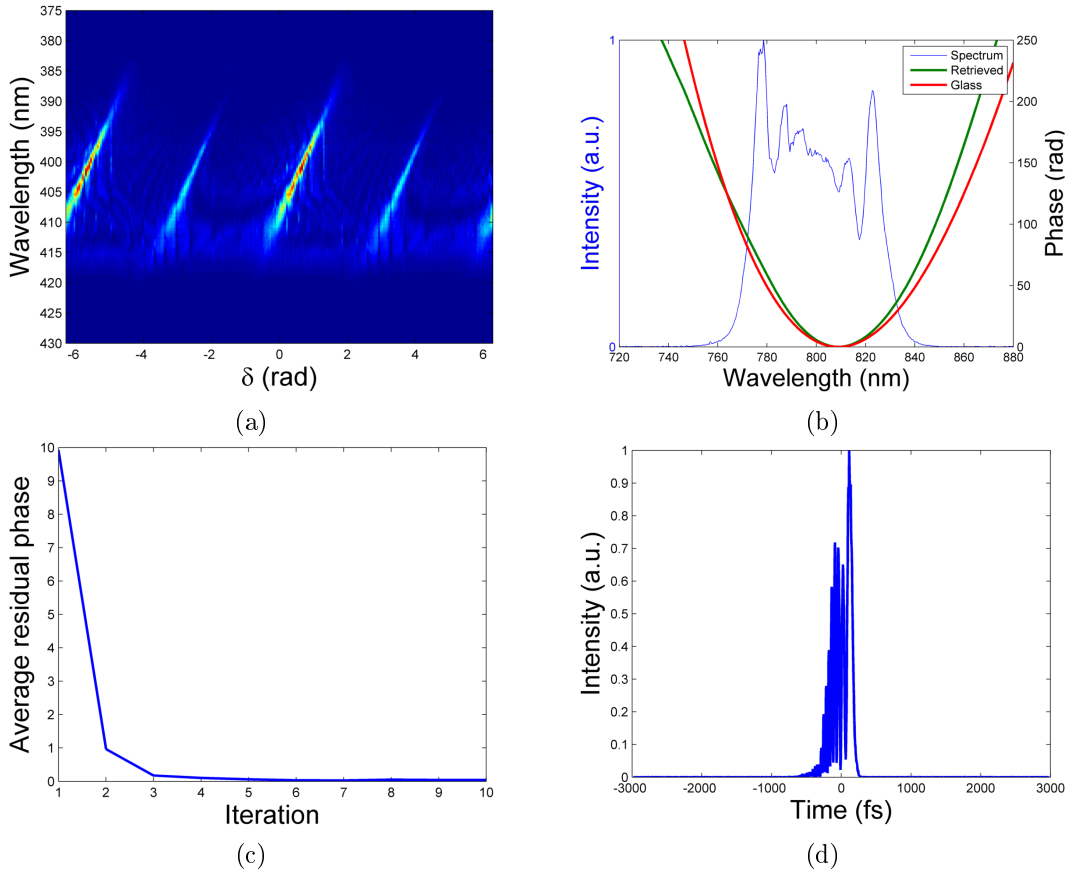


Figure 5.12: Results obtained after 10 MIIPS iterations of compressed pulse through 2 cm x 2 cm x 2 cm SF6 glass cube with a GVD of 327.68 fs^2/mm[3]. (a) MIIPS trace after 10^{th} iteration with phase function parameters (20,20). (b) Total retrieved phase (green), Calculated phase (red) superimposed on the spectrum (blue). (c) Average residual phase after each iteration. (d) Representation of pulse with total phase subtracted from glass phase.

After about 3 iterations the majority of the phase that MIIPS can retrieve, with the chosen phase function parameters, has been retrieved (figure 5.12b). The total phase retrieved after the 10^{th} scan, as illustrated by the green curve in figure(5.12b), is more than the added phase from the glass cube, as illustrated by the red curve. Computing a fit of the total retrieved phase reveal that the retrieved GVD is ~ 388 fs^2/mm, an 18% increase from the glass GVD. One can conclude from this result that MIIPS is indeed able to retrieve the phase of supercontinuum pulses. This measurement can however not say to what accuracy MIIPS is able to retrieve the phase.

The retrieved total phase being larger than the phase distortions caused by the glass cube can possibly be attributed to the following reasons:

1. The phase of the input pulse was not adequately determined in the scans prior to adding the glass cube.

2. The glass cube was not placed perfectly perpendicular to the beam.

If it were the case that MIIPS did not compress the pulses prior to adding the glass cube, then one would expect the retrieved phase after adding the glass cube to be less

than the calculated phase. Under the same conditions, with regards to phase function parameters, one would expect the error in retrieval to be similar for both scans. From this one assumes that for both scans that MIIPS will retrieve less phase. This is clearly not the case.

As stated earlier, placing the glass cube in another configuration that is not perpendicular to the incoming beam changes the beam path. This can be explained with Snell's law and also implies that the beam has travelled through more than 20 mm of glass adding more phase to the supercontinuum pulse, and changing the accuracy of estimating the phase from the glass cube.

Optical components placed after the glass cube will not have the same effect as before the beam was shifted, which can also contribute to more phase distortions than before.

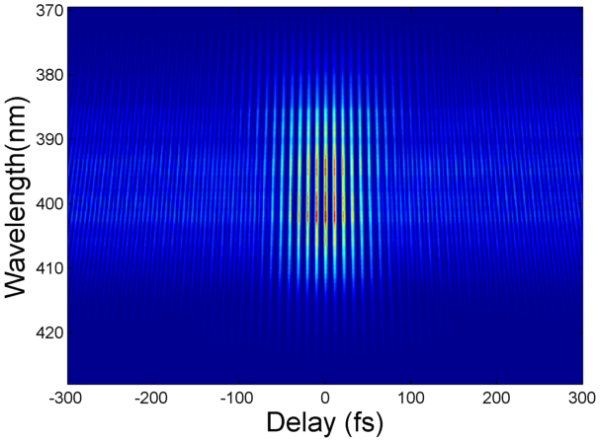
A combination of these contributions can possibly account for total retrieved phase being more than the calculated value. The structures within the MIIPS trace also indicate that something is amiss. Comparing the four structures with each other, one sees that there is a clear intensity difference between them. From experience, this unequal intensity distribution among the structures is due to the alignment of the system not being optimal. This leads one further to conclude that the additional phase retrieved is due to a beam shift caused by introducing the glass cube into the beam path at an angle that is not perpendicular to the beam.

This method has allowed us to gauge the extent to which MIIPS can reconstruct the phase of dispersed supercontinuum pulses, but has not unequivocally shown the degree of accuracy of the MIIPS procedure.

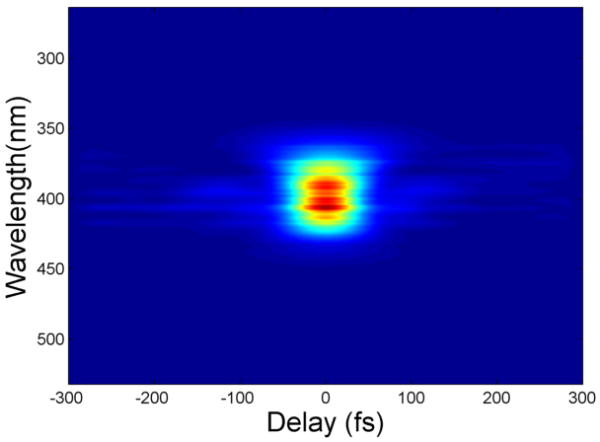
5.3.3 IFROG test

We can further gauge the validity of our measurements by performing a shaper assisted IFROG. This method is used to reconstruct the phase of an already compressed pulse. Performing this measurement on uncompressed pulse in this case is not possible, as the pulse lengths fall outside the range of pulse durations that this method can measure. In other words, the pulses are too broad for this method to measure.

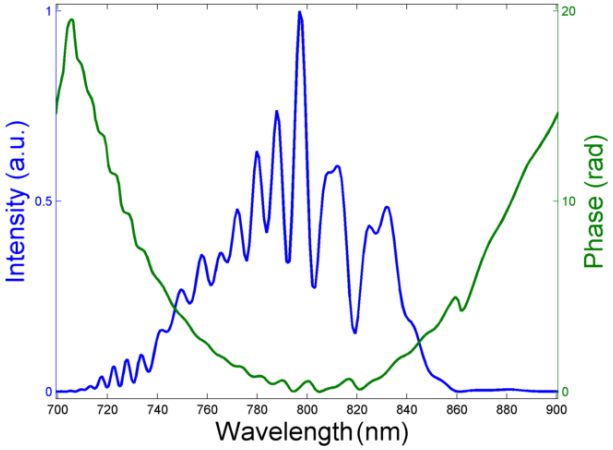
This shaper assisted IFROG is performed by programming a transfer function on the SLM that produces a double pulse with some time delay. Changing this time delay over some interval and collecting the resulting SHG signal will produce an IFROG spectrogram as shown in figure (5.13a).



(a) SLM assisted IFROG measurement after compression with MIIPS procedure with phase function parameters $\alpha = 25$ and $\gamma = 15$.



(b) Resulting FROG trace after removing background from IFROG measurement.



(c) Reconstructed spectrum (blue curve) and phase (green curve) of supercontinuum pulses with FROG algorithm

Figure 5.13: Results obtained from performing a SLM assisted IFROG measurement on an already compressed pulses. The FROG error was reported to be 0.01 with the retrieved field having a temporal FWHM of 23.38 fs.

To analyse the IFROG data, one has to convert the data to FROG data. This is done by isolating and extracting the FROG terms in the Fourier domain as discussed in chapter 3 section 3.5. Once the FROG spectrogram is extracted, it is passed through the FROG processing algorithm to produce the results presented in figure (5.13).

Figure (5.13c) shows the reconstructed phase (green) of a compressed pulse. It would seem that large phase distortions are present, but this is not the case. The phase for large portions of the spectrum are close to zero, while moving towards the edges of the spectrum it increases. Even with this quadratic phase profile, the pulse length was calculated to be roughly ~ 23 fs. This value is almost four times smaller than the input pulse with a pulse length (80 fs) and orders of magnitude smaller than the picosecond pulses produced by the PCF. The transform limited pulse length is in the order of ~ 12 fs. We can assume the pulses are Fourier limited compared to the picosecond pulses produced by the PCF.

Chapter 6

Conclusion and outlook

For the aim of developing infrastructure towards nonlinear microscopies, we developed and characterised an integral part - a pulse compressor. This pulse compressor should provide the ultra short pulses required to access non linear properties of materials to be studied.

The core principle on which this system is based is the inversely proportional nature of light between the temporal and frequency domains. This, as stated before, describes that when the frequency bandwidth of light increases (and is kept orderly) then the result in the temporal domain will be that the temporal pulse duration will become smaller.

An all-normal dispersion photonic crystal fibre (ANDi PCF) was pumped by an 80 fs pump pulse with a frequency bandwidth of about 12 nm. The resulting light from the PCF was a broadened frequency bandwidth, with the broadening dependent on the average input pulse power. Optimal operating parameters were established by studying the polarisation characteristics of the output beam. With these characteristics known, one is able to easily implement the PCF in the pulse compressor.

Characterisation of the PCF output, carried out by performing MIIPS measurements with varying MIIPS parameters, yielded promising results. Results from experimentally performed MIIPS matched those from simulations and indicated that for PCF input powers of 100 mW, translating to 100 nm bandwidth, supercontinuum pulses can be and were compressed.

Accompanying measurements strengthened the assertion that the pulse compressor, under the conditions stated in the results, is able to compress supercontinuum pulses produced by the ANDi PCF.

A comparative study between MIIPS, GMIIPS and SSMIIPS as future endeavours can be conducted, considering that the theoretical background for the GMIIPS and SSMIIPS techniques have been investigated. This study could focus on the accuracy of phase retrieval, speed of retrieval specifically applied to the output from the ANDi PCF - in order to determine which of these is most suitable for the pulse compressor.

For this study we worked with pulses that we were confident had reasonable phase

distortions that the SLM would be able to accommodate. Ideally one wants to be able to utilise the PCF to the maximum, i.e. the broadest output it can produce in this system. In working towards this one, has to do some pre-compression. This can be done by introducing a prism pulse compressor. This is a possibly good solution, as we are aware that the PCF output has mostly quadratic phase distortions, which is exactly what the prism pulse compressor corrects for.

Notwithstanding considerations for future improvements, the pulse compressor in its current state is able to compress supercontinuum light to near the transform limit and can be implemented in a microscopy setup.

Appendices

Appendix A

Experimental setup alignment

Within this appendix is contained some of the procedures followed to align certain experimental setups.

A.1 IAC

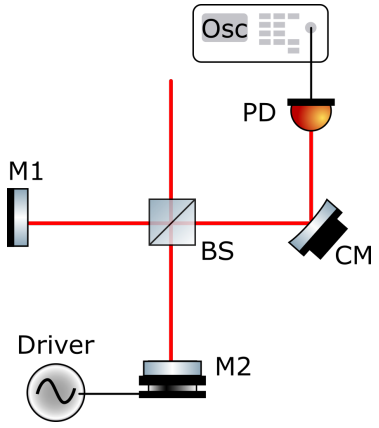


Figure A.1: Schematic representation of an interferometric autocorrelator consisting of mirrors (M1 and M2 mounted on a speaker); a curved mirror (CM); a beam splitter (BS) 45^o to incoming beam; a signal generator (driver); a two photon detector (PD) and an oscilloscope (Osc).

To align this system the following steps are followed:

1. Align beam parallel to optical table with two separated pinholes set to equal height.
2. Place BS at 45 degrees to incoming beam.
3. Place mirrors approximately as indicated in figure 4.2
4. To initially align these mirrors one can close one of the pinholes until it just lets the beam through and look at the reflection of the beams on the pinhole. If the power is too low an infra-red viewer can be used to see the reflection. Align the mirrors

so that the reflection propagates through the pinhole again. This is done for each beam separately by blocking one beam, aligning it and then switching to the other.

5. Place a curved mirror as indicated and the place PD in the focus of the mirror.

6. Connect PD to the oscilloscope and the speaker to the signal generator and switch speaker on.

7. To find the IAC signal one iteratively changes the distance of M1 and adjusts the mirrors' alignment, optimising for signal intensity on the oscilloscope until an IAC signal can be seen.

8. When the signal is found the scale on the oscilloscope can be decreased in order to observe multiple IAC signals simultaneously. The overlap distance should be adjusted such that these signals are equidistant with respect to each other.

9. Alignment is complete when background to peak ratio is approximately 1:8.

A.2 4f-Shaper

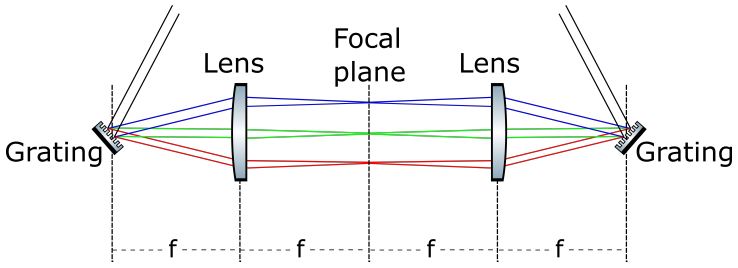


Figure A.2: 4f Shaper represented illustrates ability to separate light into its spectral components and mapping each component to a spatial coordinate in the focal plane.

1. Place a mirror where grating G1 should be placed and change alignment of mirrors M5 and M6 to propagate the beam through pinhole I to this temporary mirror. The angle of incidence relative to G1 is determined by the chosen grating.

2. Align temporary mirror such that the beam is parallel to the optical table and along the centre of where the system is to be set up.

3. Lens L2 is then to be placed a distance 2f away from where L1 is to be placed with its flat side facing the incoming beam. Monitor the reflected light to align it perpendicular on the beam. Ensure the beam propagates through the centre of the lens.

4. Place lens L1 at the appropriate position with its curved side facing the incoming beam. Optimise position such that beam propagates through the centre of the lens and perpendicular to the beam. Monitor reflection to optimise.

5. Add blazed grating G1 to the system rotating it such that chosen optimised diffraction order is in the direction of L1 and a more or less flat wave front in the focus of L1. Keep beam parallel to optical table.

6. Add G2 to the system as indicated in figure (4.3). G2 should be in the focus of L2 and rotated such that recombined beam propagates through pinhole I.

Appendix B

Matlab code

Within this appendix is presented the Matlab code to perform MIIPS, GMIIPS, and SSMIIPS simulations. These simulations are able to accept fully characterised temporal PCF outputs (amplitude and phase) or measured spectra with manually added phase estimations.

B.1 MIIPS Simulation

```
1  function [wn,df,dt] = simulate_miips(spec,n)
2 %for n==1
3 %input spectrum from spectrometer, converts and linearises to
      frequency
4 %for n==2
5 %input full characterized pulse amplitude and phase - time
      domain
6
7 c = 2.998*10^8;
8 wc =800e-9;
9 fc = 2*pi*c/wc;
10
11 if n==1
12
13 int1 = sqrt(spec(:,1));
14 i = find(int1/max(int1) > 0.063);
15
16 int2 = int1(i(1):i(length(i)));
17
18 wexp1 = spec(:,2)*10^-9;
19 wexp2 = wexp1(i(1):i(length(i)));
```

```
20
21 wc =(wexp2(ceil(length(wexp2)/2)));
22 fc = 2*pi*c/(wc);
23
24 fexp = 2*pi*c./wexp2;
25 wexpn_len = length(int2);
26 figure(2)
27 plot(fexp,int2);
28
29 wn = 2^(ceil(log2(wexpn_len))+1);
30 dw = abs(fexp(1)-fexp(length(fexp)))/wn ;
31 dt = 2*pi/(wn*dw);
32 tn = wn;
33 t = dt*5*linspace(-tn/2,tn/2-1,tn*5);
34 f = dw*linspace(-tn/2,tn/2-1,tn);
35 spec = interp1(fexp,int2,fc+f,'spline',0);
36 figure(1);
37 plot(f,spec);
38 w=f;
39 figure(2);
40 plot( t,abs(fftshift(fft(spec,5*length(spec)))));
41 xlabel('Time (s)','FontSize',12);
42 ylabel('Intensity (a.u)','FontSize',12);
43 end;
44
45 if n ==2
46     int = spec(1:length(spec),1);
47     int2 = fftshift((fft(fftshift(int))));
48     t = spec(1:length(spec),2);
49
50     w_m = fftshift(wspace(t));%Creates frequency axis
51
52     wn = 2^(ceil(log2(length(int))));
53     tn=wn;
54     dw = abs(max(w_m) - min(w_m))/wn;
55     dt = 2*pi/(wn*dw);
56     w = dw*linspace(-wn/2,wn/2+1,wn);
57     t = dt*linspace(-tn/2,tn/2-1,tn);
58     spec = interp1(w_m,int2,w,'spline',0);
```

```
59
60 end;
61
62 tau = 15e-15;
63 A =25;
64 iter = 10;
65 comp_phase_new = zeros(1,length(w));
66
67 err=0;
68
69 for count = 1:iter
70
71
72     [phi_ax,phi_out1,phi_out2,w_out,~,~,p_1,p_2,l_1,l_2] = scan(
       w,spec,comp_phase_new,A,tau,n);
73
74     figure(3);
75     set(gcf,'color','w');
76     subplot(2,2,1);
77     imagesc(phi_ax,w,w_out');
78     axis([0 4*pi -6e14 5e14]);
79     xlabel('\delta (rad)','FontSize',18);
80     ylabel('Frequency (rad/s)','FontSize',18);
81     set(gca,'xtick',0:pi:4*pi,'fontsize',14);
82     set(gca,'xticklabel',{'0','pi','2pi','3pi','4pi'});
83
84     f11 = A*tau^2*(sin(tau*w - phi_out1) + sin(tau*w - phi_out2)
       )/2;
85
86     phase = intt(intt(dw^2*f11,w),w);
87     comp_phase_new = comp_phase_new + phase;
88
89     err = plotts(comp_phase_new,phase,spec,w,dw,count,err);
90 end;
91
92 function [phi_ax,phi_out1,phi_out2,w_out,q_phase,qu_phase,p_1,
   p_2,l_1,l_2] = scan(w,spec,comp_phase,A,tau,n)
93     q_phase = 6.5536e-27;
94     qu_phase =0;%100000e-45;
```

```
95      quad_phase =0;% 1000e-60;
96      phi_ax = linspace(0,4*pi,256);
97      phi_ax = phi_ax(1:length(phi_ax));
98      count = 0;
99      if n==1
100     for phi = phi_ax;
101         count = count + 1;
102         w_out(count,:) = (abs(fftshift(fft(fftshift(fftshift(
                ifft(fftshift(spec.*exp(1i*(-comp_phase + q_phase.*w
                .^2+qu_phase.*w.^3+quad_phase.*w.^4+0*sin(w*90))).*
                exp(A*1i*sin(tau*w-phi))))).^2)))).^2);
103     end;
104     end;
105
106     if n==2
107        for phi = phi_ax;
108         count = count + 1;
109         w_out(count,:) = (abs(fftshift(fft(fftshift(fftshift(
                ifft(fftshift(spec.*exp(A*1i*sin(tau*w-phi)).*exp(-1i
                *comp_phase))))).^2)))).^2);
110     end
111     end;
112
113     wt =tau*w;
114     dp = phi_ax(2)- phi_ax(1);
115     pn = 256;
116     count2 = 0;
117     phi_out1 = zeros(1,length(w));
118     phi_out2 = zeros(1,length(w));
119     p_1 =zeros(1,length(w));
120     p_2 =zeros(1,length(w));
121     l_1 =zeros(1,length(w));
122     l_2 =zeros(1,length(w));
123
124     for i = 1:length(w);
125         count2 = count2 + 1;
126
127         p1 = int32(floor((wt(count2) + pi)/(2*dp))+1.5);
128         if(p1<1)
```

```
129             p1=1;
130         end;
131         if(p1 >pn)
132             p1 = pn;
133         end;
134         p2 = int32(floor((wt(count2) +3*pi)/(2*dp))+1.5);
135         if(p2<1)
136             p2=1;
137         end;
138         if(p2 >pn)
139             p2 = pn;
140         end;
141         l1 = p2 -p1;
142         p3 = int32(floor((wt(count2) +5*pi)/(2*dp))+1.5);
143         if(p3<1)
144             p3=1;
145         end;
146         if(p3 >pn)
147             p3 = pn;
148         end;
149         l2 = p3 -p2;
150
151         slices = w_out(:,count2);
152
153       [max1 I_1] = max(slices(p1:(p1+l1)));
154       [max2 I_2] = max(slices(p2:(p2+l2)));
155
156         phi_out1(count2) = phi_ax(I_1+p1-1);
157         phi_out2(count2) = phi_ax(I_2+p2-1);
158
159     p_1(count2)= p1;
160     p_2(count2)= p2;
161     l_1(count2)= l1;
162     l_2(count2)= l2;
163     end;
164
165 function [comp_phase] = intt(f11,w)
166 slice1 = fliplr(f11(1:(floor(length(f11)/2))));
167 slice2 = f11((floor(length(f11)/2)+1):(length(f11)-1));
```

```
168
169 comp_phase1 =cumsum(slice1);
170 comp_phase1 = -fliplr(comp_phase1);
171
172 comp_phase2 =  cumsum(slice2);
173
174 comp_phase = horzcat(comp_phase1,0,comp_phase2);
175
176 function[err]= plotts(comp_phase_new,phase,spec,w,dw,count,err)
177
178 figure(3);
179
180
181     subplot(2,2,2);
182     true_p =(unwrap(angle(spec))- max(unwrap(angle(spec))));
183     [AX,H1,H2] =plotyy(w,abs(spec).^2/max(abs(spec).^2),w,[phase
        ' comp_phase_new' true_p']);
184     set(AX(1),'YLim',[0 1])
185     set(get(AX(1),'Ylabel'),'String','Intensity (a.u.)','
        Fontsize',18);
186     set(get(AX(2),'Ylabel'),'String','Phase (rad)','Fontsize'
        ,18);
187     xlabel('Frequency (rad/s)', 'FontSize',18);
188     set(AX(2),'YLim',[-1 5],'ytick',-1:1:5)
189     set(AX(1),'XLim',[-2.5e14 2.5e14],'fontsize',14)
190     set(AX(2),'XLim',[-2.5e14 2.5e14],'fontsize',14)
191     set(H2,'LineWidth',2);
192
193     wn = 2^(ceil(log2(length(spec)))+2);
194     L = wn-length(spec);
195         spec1 = wextend('1','zpd',spec,L);
196         phase1 =wextend('1','zpd',phase,L);
197         phase2 =wextend('1','zpd',(true_p - comp_phase_new),L);
198         wn = length(spec1);
199         tn=wn;
200         dt = 2*pi/(wn*dw);
201         t = dt*linspace(-tn/2,tn/2-1,tn);
202         t= t*10^15;
203
```

```
204
205     subplot(2,2,3);
206     int_t2 =abs(fftshift(ifft(fftshift(abs(spec1).*exp(1i.*
            phase2))))).^2;
207     int_t2 = int_t2/max(int_t2);
208     t_adj = t(find(int_t2==1));
209     plot(t-t_adj,int_t2,'LineWidth',2);
210     xlabel('Time (fs)', 'FontSize',18);
211     ylabel('Intensity (a.u.)', 'FontSize',18);
212     axis([-20 20 0 max(int_t2)]);
213     set(gca,'fontsize',14);
214
215     subplot(2,2,4);
216
217     i = find(abs(spec)/max(abs(spec)) > 0.05);
218     phase2 = phase(i(1):i(length(i)));
219     nr = length(phase2);
220     err(count) = sum(abs(phase2))/nr;
221     plot(err,'LineWidth',2);
222     xlabel('Iteration', 'FontSize',18);
223     ylabel('Average residual phase', 'FontSize',18);
224     set(gca,'fontsize',14,'xtick',1:1:10);
```

B.2 GMIIPS Simulation

```
function [wn,df,dt] = simulate_gmiips(spec1,n,gate)
c = 2.998*10^8;
wc =800e-9;
fc = 2*pi*c/wc;


if n==1

int1 = sqrt(spec(:,1));
i = find(int1/max(int1) > 0.0555);

int2 = int1(i(1):i(length(i)));

wexp1 = spec(:,2)*10^-9;
wexp2 = wexp1(i(1):i(length(i)));

wc =(wexp2(ceil(length(wexp2)/2)));
fc = 2*pi*c/(wc);

fexp = 2*pi*c./wexp2;
wexpn_len = length(int2);
figure(2)
plot(fexp,int2);

wn = 2^(ceil(log2(wexpn_len))+1);
dw = abs(fexp(1)-fexp(length(fexp)))/wn ;
dt = 2*pi/(wn*dw);
tn = wn;
t = dt*5*linspace(-tn/2,tn/2-1,tn*5);
f = dw*linspace(-tn/2,tn/2-1,tn);
spec = interp1(fexp,int2,fc+f,'spline',0);
figure(1);
plot(f,spec);
w=f;
end;

if n ==2
```

```
38      int = spec1(1:length(spec1),1);
39      int2 = fftshift((fft(fftshift(int))));
40      t = spec1(1:length(spec1),2);
41
42     w_m = fftshift(wspace(t));
43
44
45      wn = 2^(ceil (log2(length(int))));
46      tn=wn;
47      dw = abs(max(w_m) - min(w_m))/wn;
48      dt = 2*pi/(wn*dw);
49      w = dw*linspace(-wn/2,wn/2+1,wn);
50      t = dt*linspace(-tn/2,tn/2-1,tn);
51      spec = interp1(w_m,int2,w,'spline',0);
52
53
54
55       figure(1);
56       plot(t,abs(int).^2./(max(abs(int).^2)));
57          axis([-2e-12 3e-12 0 1])
58          xlabel('Time (s)','Fontsize',14)
59          ylabel('Intensity (a.u.)','Fontsize',14)
60
61       figure(2);
62       plot(w+fc,abs(spec).^2);
63            xlabel('Frequency (rad/s)','Fontsize',14)
64            ylabel('Intensity (a.u.)','Fontsize',14)
65
66      a=0;
67  end;
68
69  tau = 10e-15;
70  A =25;
71  iter = 10;
72  sig = gate;
73  comp_phase_new = zeros(1,length(w));
74  error1 = zeros([length(w) iter]);
75  prev_phase=0;
76  err=0;
```

```
77
78 for count = 1:iter
79
80     [phi_ax,phi_out,w_out,~,~] = scan(w,spec,comp_phase_new,A,
       tau,n,sig);
81
82     figure(3);
83     set(gcf,'color','w');
84     subplot(2,2,1);
85     imagesc(phi_ax,w,w_out');
86     xlabel('\delta (rad)','FontSize',14);
87     ylabel('Frequency (rad/s)','FontSize',14);
88     set(gca,'xtick',-2*pi:pi:2*pi);
89     set(gca,'xticklabel',{'-2pi','-pi','0','pi','2pi'});
90     axis([-2*pi 2*pi -6e14 4e14])
91
92    dw = abs(w(2)-w(1));
93     f11 = dw^2*A*tau^2*(sin(tau*w - phi_out));
94
95     phase = intt(intt(f11,w),w);
96     comp_phase_new = comp_phase_new + phase;
97
98
99     err=  plotts(comp_phase_new,phase,spec,w,dw,count,err);
100
101 end;
102
103 function [phi_ax,phi_out,w_out,q_phase,qu_phase] = scan(w,spec,
    comp_phase,A,tau,n,sig)
104     q_phase = 1000e-30;
105     qu_phase =100000e-45;
106     quad_phase = 1000e-60;
107     phi_ax = linspace(-2*pi,2*pi,512);
108     phi_ax = phi_ax(1:length(phi_ax));
109     count = 0;
110     w_out=zeros(512,length(w));
111     if n==1
112     for phi = phi_ax;
113         count = count + 1;
```

```
114         w_out(count,:) = (abs(fftshift(fft(fftshift(fftshift(
                ifft(fftshift(spec.*exp(1i*(-comp_phase + q_phase.*w
                .^2+qu_phase.*w.^3+quad_phase.*w.^4+0*sin(w*90))).*
                exp(-((tau.*w - phi)/sig).^2 +A*1i*cos(tau*w-phi)))))
                .^2)))).^2);
115
116     end;
117     end;
118
119     if n==2
120         for phi = phi_ax;
121          count = count + 1;
122          w_out(count,:) = (abs(fftshift(fft(fftshift(fftshift(
                ifft(fftshift(spec.*exp( A*1i*sin(tau*w-phi)).*exp
                (-((tau.*w - phi)./sig).^2).*exp(-1i*comp_phase))))
                .^2)))).^2);
123         end
124     end;
125
126     phi_out = zeros(1,length(w));
127
128     for i = 1:length(w);
129         [~, I1]=max(w_out(:,i));
130         phi_out(i) = phi_ax(I1);
131     end;
132
133 function [comp_phase] = intt(f11,w)
134 slice1 = fliplr(f11(1:(floor(length(f11)/2))));
135 slice2 = f11((floor(length(f11)/2)+1):(length(f11)-1));
136
137 comp_phase1 =cumsum(slice1);
138 comp_phase1 = -fliplr(comp_phase1);
139
140 comp_phase2 =  cumsum(slice2);
141
142 comp_phase = horzcat(comp_phase1,0,comp_phase2);
143 a=0;
144
145 function [err] = plotts(comp_phase_new,phase,spec,w,dw,count,err
```

```
    )

figure(3);


    subplot(2,2,2);
    true_p =(unwrap(angle(spec))- max(unwrap(angle(spec))));
    [AX,H1,H2] =plotyy(w,abs(spec).^2/max(abs(spec).^2),w,[phase
        ' comp_phase_new' true_p']);
    set(AX(1),'YLim',[0 1])
    set(get(AX(1),'Ylabel'),'String','Intensity (a.u.)','
       Fontsize',18);
    set(get(AX(2),'Ylabel'),'String','Phase (rad)','Fontsize'
        ,18);
    xlabel('Frequency (rad/s)', 'FontSize',18);
    set(AX(2),'YLim',[-150 50],'ytick',-150:50:50)
    set(AX(1),'XLim',[-6e14 4e14],'fontsize',14)
    set(AX(2),'XLim',[-6e14 4e14],'fontsize',14)
    set(H2,'LineWidth',2);

    wn = 2^(ceil(log2(length(spec)))+1);
    L = wn-length(spec);
         spec1 = wextend('1','zpd',spec,L);
         phase1 =wextend('1','zpd',phase,L);
         phase2 =wextend('1','zpd',(true_p - comp_phase_new),L);
         wn = length(spec1);
         tn=wn;
         %dw = abs(max(w_m) - min(w_m))/wn;
         dt = 2*pi/(wn*dw);
         t = dt*linspace(-tn/2,tn/2-1,tn);
         t= t*10^15;

    subplot(2,2,3);
   int_t2 =abs(fftshift(ifft(fftshift(abs(spec1).*exp(1i.*phase2
       )))))).^2;
    int_t2 = int_t2/max(int_t2);
    t_adj = t(find(int_t2==1));
    plot(t-t_adj,int_t2,'LineWidth',2);
    xlabel('Time (fs)', 'FontSize',18);
```

```
180     ylabel('Intensity (a.u.)', 'FontSize',18);
181     set(gca,'fontsize',14);
182
183     subplot(2,2,4);
184
185     i = find(abs(spec)/max(abs(spec)) > 0.05);
186     phase2 = phase(i(1):i(length(i)));
187     nr = length(phase2);
188     err(count) = abs(sum(phase2))/nr;
189     plot(err,'LineWidth',2);
190     xlabel('Iteration', 'FontSize',18);
191     ylabel('Average residual phase', 'FontSize',18);
192     set(gca,'fontsize',14,'xtick',1:1:10);
```

B.3 SSMIIPS Simulation

```
function [wn,df,dt] = simulate_miipsv2(spec,n,k)
%for n==1
%input spectrum from spectrometer, converts and linearises to
    frequency
%for n==2
%input full characterized pulse amplitude and phase - time
    domain

c = 2.998*10^8;

if n==1

int1 = sqrt(spec(:,1));
i = find(int1/max(int1) > 0.13);

int2 = int1(i(1):i(length(i)));

wexp1 = spec(:,2)*10^-9;
wexp2 = wexp1(i(1):i(length(i)));

wc =(wexp2(ceil(length(wexp2)/2)));
fc = 2*pi*c/(wc);

fexp = 2*pi*c./wexp2;
wexpn_len = length(int2);
figure(2)
plot(fexp,int2);

wn = 2^(ceil(log2(wexpn_len))+1);
dw = abs(fexp(1)-fexp(length(fexp)))/wn ;
dt = 2*pi/(wn*dw);
tn = wn;
t = dt*5*linspace(-tn/2,tn/2-1,tn*5);
f = dw*linspace(-tn/2,tn/2-1,tn);
spec = interp1(fexp,int2,fc+f,'spline',0);
figure(1);
plot(f,spec);
```

```
w=f;
figure(2);
plot( t,abs(fftshift(fft(spec,5*length(spec)))));
xlabel('Time (s)','FontSize',12);
ylabel('Intensity (a.u)','FontSize',12);
end;

if n ==2
    int = spec(1:length(spec),1);
    int2 = fftshift((fft(fftshift(int))));
    t = spec(1:length(spec),2);

    w_m = fftshift(wspace(t));

    wn = 2^(ceil (log2(length(int))));
    tn=wn;
    dw = abs(max(w_m) - min(w_m))/wn;
    dt = 2*pi/(wn*dw);
    w = dw*linspace(-wn/2,wn/2+1,wn);
    t = dt*linspace(-tn/2,tn/2-1,tn);
    spec = interp1(w_m,int2,w,'spline',0);

end;

tau = 15e-15;
A =25;

comp_phase_new = zeros(1,length(w));
err=0;
%%%%Retrieve phase from experimental
[phi_ax,phi_out1,phi_out2,w_out,~,~,p_1,p_2,l_1,l_2] = scan(w,
    spec,comp_phase_new,A,tau,n); %Retrieve phase from Mexp

dw = w(2)-w(1);
gdd_0 = A.*tau^2.*(sin(tau.*w - phi_out1) + sin(tau.*w -
    phi_out2))/2;
phase = intt(intt(dw^2.*gdd_0,w),w);%Integrate twice
```

```
73 count=1;
74
75 figure(3);
76     set(gcf,'color','w');
77     imagesc(phi_ax,w,w_out');
78     xlabel('\delta (rad)','FontSize',14);
79     ylabel('Frequency (rad/s)','FontSize',14);
80     set(gca,'xtick',-2*pi:pi:2*pi);
81     set(gca,'xticklabel',{'-2pi','-pi','0','pi','2pi'});
82     err=  plotts(comp_phase_new,phase,spec,w,dw,count,err);
83
84
85 run=true;
86 count =2;
87 %%%Computationally retrieve phase
88
89 while(run==true)
90
91     [phi_ax,phi_out1,phi_out2,w_out,~,~,~,~,~,~] = scan(w,abs(
           spec),phase,A,tau,n);
92
93     figure(4);
94     set(gcf,'color','w');
95     subplot(2,2,1);
96     imagesc(phi_ax,w,w_out');
97     xlabel('\delta (rad)','FontSize',14);
98     ylabel('Frequency (rad/s)','FontSize',14);
99     set(gca,'xtick',-2*pi:pi:2*pi);
100    set(gca,'xticklabel',{'-2pi','-pi','0','pi','2pi'});
101
102    feedback = A.*tau.^2.*(sin(tau.*w - phi_out1) + sin(tau.*w -
           phi_out2))./2;
103    a=0;
104
105    gdd_err = feedback - gdd_0;
106    gdd = gdd_0 - k.*gdd_err;
107
108    phase = intt(intt(dw^2.*gdd,w),w);
109
```

```
110      GDD_err = ((tau^2)*A)/20;
111
112      [err region] = plotts(comp_phase_new,phase,spec,w,dw,count,
             err);
113
114      count=count+1;
115  if count == 11
116
117      run = false;
118  end;
119  end;
120
121  function [phi_ax,phi_out1,phi_out2,w_out,q_phase,qu_phase,p_1,
     p_2,l_1,l_2] = scan(w,spec,comp_phase,A,tau,n)
122      q_phase = 500e-30;
123      qu_phase =10000e-45;
124      quad_phase =0;% 1000e-60;
125      phi_ax = linspace(-2*pi,2*pi,256);
126      phi_ax = phi_ax(1:length(phi_ax));
127      count = 0;
128      if n==1
129      for phi = phi_ax;
130          count = count + 1;
131          w_out(count,:) = (abs(fftshift(fft(fftshift(fftshift(
                 ifft(fftshift(spec.*exp(1i*(-comp_phase + q_phase.*w
                 .^2+qu_phase.*w.^3+quad_phase.*w.^4+0*sin(w*90))).*
                 exp(A*1i*sin(tau*w-phi))))).^2)))).^2);
132      end;
133      end;
134
135      if n==2
136          for phi = phi_ax;
137           count = count + 1;
138          w_out(count,:) = (abs(fftshift(fft(fftshift(fftshift(
                 ifft(fftshift(spec.*exp(A*1i*sin(tau*w-phi)).*exp(1i*
                 comp_phase))))).^2)))).^2);
139          end
140      end;
141
```

```
142     wt =tau*w;
143     dp = phi_ax(2)- phi_ax(1);
144     pn = 256;
145     count2 = 0;
146     phi_out1 = zeros(1,length(w));
147     phi_out2 = zeros(1,length(w));
148     p_1 =zeros(1,length(w));
149     p_2 =zeros(1,length(w));
150     l_1 =zeros(1,length(w));
151     l_2 =zeros(1,length(w));
152
153     for i = 1:length(w);
154         count2 = count2 + 1;
155
156         p1 = int32(floor((wt(count2) + pi)/(2*dp))+1.5);
157         if(p1<1)
158             p1=1;
159         end;
160         if(p1 >pn)
161             p1 = pn;
162         end;
163         p2 = int32(floor((wt(count2) +3*pi)/(2*dp))+1.5);
164         if(p2<1)
165             p2=1;
166         end;
167         if(p2 >pn)
168             p2 = pn;
169         end;
170         l1 = p2 -p1;
171         p3 = int32(floor((wt(count2) +5*pi)/(2*dp))+1.5);
172         if(p3<1)
173             p3=1;
174         end;
175         if(p3 >pn)
176             p3 = pn;
177         end;
178         l2 = p3 -p2;
179
180         slices = w_out(:,count2);
```

```
181
182         [max1 I_1] = max(slices(p1:(p1+l1)));
183         [max2 I_2] = max(slices(p2:(p2+l2)));
184
185             phi_out1(count2) = phi_ax(I_1+p1-1);
186             phi_out2(count2) = phi_ax(I_2+p2-1);
187
188       p_1(count2)= p1;
189       p_2(count2)= p2;
190       l_1(count2)= l1;
191       l_2(count2)= l2;
192       end;
193
194 function [comp_phase] = intt(f11,w)
195 slice1 = fliplr(f11(1:(floor(length(f11)/2))));
196 slice2 = f11((floor(length(f11)/2)+1):(length(f11)-1));
197
198 comp_phase1 =cumsum(slice1);
199 comp_phase1 = -fliplr(comp_phase1);
200
201 comp_phase2 =  cumsum(slice2);
202
203 comp_phase = horzcat(comp_phase1,0,comp_phase2);
204 a=0;
205
206 function[err,region]= plotts(comp_phase_new,phase,spec,w,dw,
      count,err)
207
208 figure(4);
209       subplot(2,2,2);
210       true_p =(unwrap(angle(spec))- max(unwrap(angle(spec))));
211       res_phase = true_p - phase;
212       [AX,H1,H2] =plotyy(w,abs(spec).^2/max(abs(spec).^2),w,[phase
            ' true_p' res_phase']);
213       title('Phase retrieval')
214  set(AX(1),'YLim',[0 1])
215       set(get(AX(1),'Ylabel'),'String','Intensity (a.u.)','
            Fontsize',18);
216       set(get(AX(2),'Ylabel'),'String','Phase (rad)','Fontsize'
```

```
    ,18);
set(AX(2),'YLim',[-6 6], 'ytick',-6:2:6)
set(AX(1),'XLim',[-6e14 6e14],'fontsize',14)
set(AX(2),'XLim',[-6e14 6e14],'fontsize',14)
set(H2,'LineWidth',2);

wn = 2^(ceil(log2(length(spec)))+1);
L = wn-length(spec);
     spec1 = wextend('1','zpd',spec,L);
     phase1 =wextend('1','zpd',true_p-phase,L);

     wn = length(spec1);
     tn=wn;
     dt = 2*pi/(wn*dw);
     t = dt*linspace(-tn/2,tn/2-1,tn);
t= t*10^15;

subplot(2,2,3);
int_t2 =abs(fftshift(ifft(fftshift(abs(spec1).*exp(1i.*
    phase1))))).^2;
int_t2 = int_t2/max(int_t2);
 t_adj = t(find(int_t2==1));
plot(t-t_adj,int_t2,'LineWidth',2);
xlabel('Time (fs)', 'FontSize',18);
ylabel('Intensity (a.u.)', 'FontSize',18);
set(gca,'fontsize',14);
set(gca,'fontsize',14);

subplot(2,2,4);

i = find(abs(spec).^2/max(abs(spec)).^2 > 0.1);
region = [i(1) i(length(i))];
phase2 = phase(i(1):i(length(i)));
phase3 = true_p(i(1):i(length(i)));
nr = length(phase2);
err(count) = abs(sum(phase3 - phase2))/nr;
plot(err);
 xlabel('Iteration', 'FontSize',18);
ylabel('Average residual phase', 'FontSize',18);
```

```
254         set(gca,'fontsize',14,'xtick',1:1:10);
```

Bibliography

[1] NL-1050-NEG-1 Nonlinear Photonic Crystal Fiber. `http://www.nktphotonics.com/files/files/NL-1050-NEG-1-100409.pdf`. Accessed: 2016-01-01.

[2] Spectra-Physics: Tsunami Mode-Locked Ti:Sapphire Laser User's Manual. 1999.

[3] Refractiveindex.info. `http://refractiveindex.info/?shelf=glass&book=SCHOTT-SF&page=SF6`. Accessed: 2016-12-01.

[4] Graf, B.W. and Boppart, S.A.: Multimodal In Vivo Skin Imaging with Integrated Optical Coherence and Multiphoton Microscopy. *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 18, no. 4, pp. 1280–1286, jul 2012. ISSN 1077-260X.

[5] Diels, J.-C. and Rudolph, W.: 1 - Fundamentals. In: Diels, J.-C. and Rudolph, W. (eds.), *Ultrashort Laser Pulse Phenomena (Second Edition)*, second edi edn, pp. 1–60. Academic Press, Burlington, 2006. ISBN 978-0-12-215493-5.

[6] Griffiths, D.J. and Inglefield, C.: Introduction to Electrodynamics. *American Journal of Physics*, vol. 73, no. 6, p. 574, 2005. ISSN 00029505.

[7] Rulliere, C.: *Femtosecond laser pulses.* 1998. ISBN 0387017690.

[8] Trebino, R.: *Frequency-Resolved Optical Gating: The Measurement of Ultrashort Laser Pulses*, vol. 32. Springer US, Boston, MA, 2000. ISBN 978-1-4613-5432-1.

[9] Diels, J.-C.M., Fontaine, J.J., McMichael, I.C. and Simoni, F.: Control and measurement of ultrashort pulse shapes (in amplitude and phase) with femtosecond accuracy. *Applied Optics*, vol. 24, no. 9, p. 1270, may 1985. ISSN 0003-6935.

[10] Bosman, G.W.: *Ultrashort Optical Pulse Characterization.* Masters Thesis, University of Stellenbosch, 2008.

[11] Diels, J.-C. and Rudolph, W.: 9 - Diagnostic Techniques. In: Diels, J.-C. and Rudolph, W. (eds.), *Ultrashort Laser Pulse Phenomena (Second Edition)*, second edi edn, pp. 457–489. Academic Press, Burlington, 2006. ISBN 978-0-12-215493-5.

[12] Dudley, J.M., Genty, G. and Coen, S.: Supercontinuum generation in photonic crystal fiber. *Reviews of Modern Physics*, vol. 78, no. 4, pp. 1135–1184, 2006. ISSN 00346861.

[13] Knight, J.: Photonic crystal fibres. *Nature*, vol. 424, no. 6950, pp. 847–851, 2003. ISSN 1476-4687.

[14] Sukhoivanov, I.A., Iakushev, S.O., Shulika, O.V., Andrade-Lucio, J.A., Díez, A. and Andrés, M.: Supercontinuum generation at 800 nm in all-normal dispersion photonic crystal fiber. *Optics Express*, vol. 22, no. 24, p. 30234, 2014. ISSN 1094-4087.

[15] Heidt, A.M.: *Novel coherent supercontinuum light sources based on all-normal dispersion fibers Dissertation*. Ph.D. thesis, 2011.

[16] Heidt, A.M.: Simulations courtesy of A. Heidt. 2016.

[17] Lozovoy, V.V., Pastirk, I. and Dantus, M.: Multiphoton intrapulse interference. IV. Ultrashort laser pulse spectral phase characterization and compensation. *Optics letters*, vol. 29, no. 7, pp. 775–7, apr 2004. ISSN 0146-9592.

[18] Comin, A., Ciesielski, R., Piredda, G., Donkers, K. and Hartschuh, A.: Compression of ultrashort laser pulses via gated multiphoton intrapulse interference phase scans. *Journal of the Optical Society of America B*, vol. 31, no. 5, p. 1118, may 2014. ISSN 0740-3224. 1401.5952.

[19] Comin, A., Ciesielski, R., Coca-López, N. and Hartschuh, A.: Phase retrieval of ultrashort laser pulses using a MIIPS algorithm. *Optics Express*, vol. 24, no. 3, p. 2505, feb 2016. ISSN 1094-4087.

[20] Stibenz, G. and Steinmeyer, G.: Structures of interferometric frequency-resolved optical gating. *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 12, no. 2, pp. 286–296, mar 2006. ISSN 1077-260X.

[21] Spectra-Physics: Millennia Vs Diode-pumped, cw Visible Laser User's Manual. 2001.

[22] Thorlabs: Guide To and Polishing. 2006.

[23] Wefers, M.M. and Nelson, K.a.: Generation of high-fidelity programmable ultrafast optical waveforms. *Optics letters*, vol. 20, no. 9, p. 1047, 1995. ISSN 0146-9592.

[24] Iaconis, C. and Walmsley, I.A.: Spectral phase interferometry for direct electric-field reconstruction of ultrashort optical pulses. *Optics Letters*, vol. 23, no. 10, p. 792, may 1998. ISSN 0146-9592.